THE DISCRETE ADJOINT APPROACH TO AERODYNAMIC SHAPE OPTIMIZATION

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF AERONAUTICS AND ASTRONAUTICS
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Siva Kumaran Nadarajah
January 2003

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Antony Jameson
(Principal Adviser)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Juan Alonso

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Robert MacCormack

Approved for the University Committee on Graduate Studies:

To My Parents,
T.C. Nadarajah and Sushila

Abstract

A viscous discrete adjoint approach to automatic aerodynamic shape optimization is developed, and the merits of the viscous discrete and continuous adjoint approaches are discussed. The viscous discrete and continuous adjoint gradients for inverse design and drag minimization cost functions are compared with finite-difference and complex-step gradients. The optimization of airfoils in two-dimensional flow for inverse design and drag minimization is illustrated. Both the discrete and continuous adjoint methods are used to formulate two new design problems. First, the time-dependent optimal design problem is established, and both the time accurate discrete and continuous adjoint equations are derived. An application to the reduction of the time-averaged drag coefficient while maintaining time-averaged lift and thickness distribution of a pitching airfoil in transonic flow is demonstrated. Second, the remote inverse design problem is formulated. The optimization of a three-dimensional biconvex wing in supersonic flow verifies the feasibility to reduce the near field pressure peak. Coupled drag minimization and remote inverse design cases produce wings with a lower drag and a reduced near field peak pressure signature.

Acknowledgments

I would like to express gratitude to my mentor, Professor Antony Jameson, whose expertise, understanding, and patience added considerably to my graduate experience. His vast knowledge and enthusiasm have inspired me greatly, and I thank him for his time and support.

A very special thanks to Dr. Juan Alonso. He provided me with direction, technical support and became a mentor and friend. I would like to thank the other members of my committee, Dr. Ilan Kroo, Dr. Robert MacCormack, and Dr. Michael Saunders for the assistance they provided at all levels of the research project. This research would not have been possible without the financial assistance of the Airforce Office of Scientific Research (AFOSR Grant No: F49620-98-1-022), Defense Advanced Research Project Agency (DARPA Grant No: MDA972-01-2-0003), Advanced Simulation and Computing: Department of Energy (ASC Grant No: B341491), and Mark Yong.

I have enjoyed the encouragement and warm company of my colleagues in the Aerospace Computing Laboratory. A special thanks to Sangho Kim for the fruitful conversations about optimization and to Matthew McMullen for the many discussions on research. Matt's humor was a wonderful distraction from work.

Life in the San Francisco Bay Area would have been meaningless without the love and friendship of the Sai Bay Area Young Adults. I cherish the hours of serving the community with such kind and loving friends. I am grateful to my Guru Sri Sathya Sai Baba for inspiring me to believe in high ideals and to live up to what I believe in. My dream is to make my life Your message.

I am eternally grateful to my parents, T.C. Nadarajah and Sushila Nadarajah,

and my family for their unconditional love. Thanks for encouraging me to be an independent thinker and having confidence in my abilities to go after new things that inspired me.

Above all, I thank Lisa Steffen, my wife and best friend, for her unbounded love and confidence in me. Lisa gives me something to look forward to each and every day.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Engineers continually strive to improve their designs, both to increase their operational effectiveness and their market appeal. While some qualities such as aesthetics are hard to measure, the factors contributing to operational performance and cost are generally amenable to quantitative analysis. In the absence of an optimization approach, crucial decisions during a design process that could ultimately affect the efficiency of a system are often left to the judgment of experienced personnel, researchers, and engineers. Since these decisions ultimately determine whether a company fails or prospers, the introduction of quantitative optimization methods can be crucial to improving its competitive strength. In the design of a complex engineering system, relatively small design changes can sometimes lead to significant benefits. For example, small changes in wing section shapes can lead to large reductions in shock strength in transonic flow. Changes of this type are unlikely to be discovered by trial and error methods, and it is in this situation that optimization methods can play a particularly important role.

Beightler, in Foundations of Optimization, [6] lists three important steps on how to optimize a system: first, understand the system and the various variables that influence it; second, decide on a measure of effectiveness that depends on the system variables that have a great influence on the efficiency of the system; third, choose those values of system variables that produce the optimum system.

In the first step, the knowledge of the inner workings of the system provides the

engineer with the fundamentals of how various variables interact with each other to influence the system. In an aircraft design group that consists of a wide variety of sub-groups such as computational fluid dynamics, stability and control, aeroelastic and flutter analysis, engineers working in a particular group should ideally have a general overview of the research and development conducted in other groups. Knowledge of the various design parameters from the numerous sub-groups and its influence on the performance of the aircraft would help them to formulate a better design problem.

The second step requires the designer to define a measure of the system effectiveness. In aircraft design, there are many figures of merit that measure the overall performance of the aircraft. The cruise lift-to-drag ratio, $\frac{L}{D}$, is particularly important, since it provides an indication of the efficiency of the aircraft. An aircraft with high $\frac{L}{D}$ either produces a large lift load or low drag. High lift capability allows an aircraft in cruise to carry a larger payload. Low drag translates to low fuel consumption and ultimately maximizes the aircraft range. The choice for a figure of merit depends on the mission of the aircraft.

In the last step, the designer proceeds to apply an optimization algorithm to produce an optimum result that satisfies the constraints of the problem. There are various optimization algorithms that one can choose from. The choice is highly dependent on the type of problem: linear or nonlinear, number of design variables, number of figures of merit, and whether the problem is unconstrained or constrained. A desire to increase $\frac{L}{D}$ without constraints can lead to an increase in the aspect ratio and thus produce an aircraft with a very large span for a given wing area, which would in turn increase the wing weight. Therefore constraints can be as important as the figure of merit in optimization.

In the last hundred years, aircraft designers have employed various methods to arrive at their final designs. The current design process in a typical aircraft design company follows three steps. First, the conceptual design group proposes new configurations in anticipation of new market requirements. The group generally is composed of twenty to thirty people and works on an ongoing basis to create new concepts and configurations. Radical new designs are predominantly presented to the market at this stage. However, most current designs are based on past designs that have been

successful in the market. Second, the preliminary design stage requires designers to meet aeroelastic, flutter, stability and control, and other performance criteria using basic empirical methods and low-fidelity simulation codes. Approximately three hundred people require two to three years and consume an estimated $\$200 - 300$ million. This estimate is based on the design of a medium commercial jet transport. A complete external configuration and major load and stresses are determined at the end of this stage. The aircraft performance is also frozen at this stage and orders from airline companies follow thereafter. Third, the detailed design stage involves the many different sub-groups manufacturing and testing the various components of the aircraft. Information is transfered between the groups to satisfy the overall system requirements. Approximately three to five thousand personnel work on arriving at the design goals guaranteed to airline companies during the preliminary design stage. This stage normally requires around $\$5 - 12$ billion and spans between three to four years.

During the preliminary and detailed design stages, large amounts of wind tunnel data are collected to improve the existing design until a satisfactory design is obtained within the scheduled time. Such a design process does not allow for vast numbers of design iterations or variables to be considered. With the introduction of computational methods, researchers in the last thirty years have used these powerful tools to provide a greater understanding of the problem and their ability to provide analysis for a greater number of designs. The ultimate aim of both the traditional and computational approaches was to improve the design within a predetermined schedule but it was still left to the design group to make the important decisions that would ultimately optimize their design.

This raises the need to introduce optimization theory into aircraft design. Historically, optimization theory has its roots in the Renaissance period. During this era the solution to the *Brachistochrone* problem attracted many great philosophers and thinkers. Galileo guessed that the circular arc would provide the shortest time for an object to travel from an elevated point A to the ground at point B. Then, in 1694, Johann Bernoulli proved mathematically that the optimum shape was in fact a cycloid.

In the past, aircraft design techniques have depended on analytical solutions to arrive at optimum shapes. Examples include the optimum lift distribution of monoplane wings, and the Sears-Haack solution for the minimum wave drag of a body of revolution in supersonic flow. In 1945 Lighthill [50] first employed the method of conformal mapping to design two-dimensional airfoils to achieve a desired target pressure distribution. These methods were restricted to incompressible flow, but later McFadden [56] extended the method to the compressible flow regime. Bauer et al. [5] and Garabedian et al. [21] established an alternate method based on complex characteristics to solve the potential equations in the hodograph plane. This method successfully produced shock-free transonic flows.

Constrained optimization for aerodynamic design was initially explored by Hicks et al. [27]. They used the finite-difference method to evaluate sensitivities. Since then optimization techniques for the design of aerospace vehicles have generally used gradient-based methods. Through the mathematical theory for control systems governed by partial differential equations established by Lions et al. [51], Pironneau et al. [68] created a framework for the formulation of elliptic design problems. The approach significantly lowers the computational cost and is clearly an improvement over classical finite-difference methods. Using control theory the gradient is calculated indirectly by solving an adjoint equation. Although there is the additional overhead of solving the adjoint equation, once it has been solved the cost of obtaining the derivatives of the cost function with respect to each design variable is negligible. Consequently, the total cost to obtain these gradients is independent of the number of design variables and amounts to the cost of one flow solution and one adjoint solution. The adjoint problem is a linear partial differential equation of lower complexity than the flow solver. Jameson was the first to apply control theory for transonic design problems [33, 34, 35]. Subsequently, Jameson et al. [37, 41, 42] pioneered the shape optimization method for Euler and Navier-Stokes problems. Automatic aerodynamic design of aircraft configurations has yielded optimized solutions of wing and wing-body configurations by Reuther et al. [69, 71] and Burgreen et al. [10].

The injection of optimization theory into the design process and innovations in computer technology have allowed researchers to attempt more complex problems.

This launched a snow-ball effect, where bigger and faster computers have allowed engineers to tackle more sophisticated systems, which then ultimately required even faster computers. Future optimization techniques will allow designers to reinvent the design process. The traditional approach discussed earlier will be replaced with a multidisciplinary approach, where the various disciplines of the design process will be coupled to allow changes in the configuration during the design process, a feature not allowed in the current approach. Low-fidelity flow models coupled with optimization techniques that may be able to identify global minimum such as genetic algorithms will be used during the conceptual and preliminary design stages. Higher fidelity and gradient-based methods with multidisciplinary capabilities will be used during the detailed design stage. This will allow the designer to determine accurately the optimal values of parameters such as the wing thickness that affect both the aerodynamics and structural performance of the aircraft. Designers will be able to examine a larger number of design cycles and this will allow them to tackle more multidisciplinary problems and perhaps arrive at radically new designs.

Optimization will enhance the designers' ability to produce a better system instead of reducing their role in the design process. Moreover, the designers' productivity will increase with the means to explore new approaches and designs. Often the sensitivity of the figure of merit with respect to the system variables is as important as the optimum result itself. Knowledge of the sensitivities provides the designer with new insights on how various system variables affect the performance of the system. This allows the designer to understand the system better and devise better problem statements to tackle the issues at hand.

The potential benefit of using optimization theory has only been realized recently with the advent of faster methods to obtain the gradient. The control theory approach to shape optimization has revolutionized the concept of utilizing computational fluid dynamics as a design tool. The ability to obtain gradients cheaply has allowed researchers to attempt new problems in aircraft design.

This thesis contributes to the development of the discrete and continuous adjoint approaches, examining three distinctively separate problems of current interest.

Continuous Versus Discrete Adjoint. The motivation for the development

of the discrete viscous adjoint equations is to study the trade-offs between the complexity of formulating the discrete adjoint equation and the accuracy of the resulting estimate of the gradient when compared to the continuous adjoint approach. The goal of this research is to evaluate both approaches for inverse design and drag minimization problems. The gradients from each approach are compared to gradients acquired using the finite-difference and complex-step methods.

Optimum Shape Design for Unsteady Flows. Helicopter rotors and turbomachinery blades operate in unsteady flow. The forward flight speed of a helicopter is restricted by the retreating blade stall limit and the advancing blade compressibility limit. In turbomachinery the flutter and stall boundaries limit the operational efficiency. It is apparent that the development of optimum design methods for unsteady flows is fundamental to improving the performance of a variety of aerospace systems. The unsteady adjoint equations are developed in this work. They are applied to the design of airfoils undergoing a pitching oscillation to reduce the time-averaged drag coefficient while maintaining the time-averaged lift coefficient.

Remote Inverse Design. A major barrier to the development of supersonic business jets is the sonic boom. A new approach to tailor the aircraft shape to minimize the sonic boom signature is developed in this research. The application of the method is aimed at modifications of the near field pressure signature of two-dimensional airfoils and wings in three-dimensional flow. A coupled remote inverse design and drag minimization objective function is also used to tailor the near field signature and control the wave drag of three-dimensional wings.

The following sub-sections develop the motivation in greater depth for each of the research goals. A description of the problem statement and the challenges faced in each area of study are discussed. In the chapters to follow a detailed study of the formulation of the equations and the results of the test cases are described and illustrated. The conclusions discuss the potential role of these methods in the future design of complex systems.

1.1 The Discrete and Continuous Adjoint Approaches

There are two approaches to develop the adjoint equations: continuous or discrete. In the continuous adjoint approach the control theory is applied to the differential equations governing the flow. The variation of the cost function and field equations with respect to the flow field variables and design variables are combined through the use of Lagrange multipliers, also called costate or adjoint variables. Collecting the terms associated with the variation of the flow field variables produces the adjoint equation and its boundary condition. The terms associated with the variation of the design variables produce the gradient. The field equations and the adjoint equation with its boundary conditions must finally be discretized to obtain numerical solutions.

In the discrete adjoint approach, control theory is applied directly to the set of discrete field equations. The discrete adjoint equation is derived by collecting together all the terms multiplied by the variation $\delta w_{i,j}$ of the discrete flow variables. If the discrete adjoint equation is solved exactly, then the resulting solution for the Lagrange multipliers produces an exact gradient of the discrete cost function, and the derivatives should be exactly consistent with gradients obtained by the complex-step method.

The discrete and continuous approaches have been pursued by a number of researchers using a wide variety of schemes and methods. Shubin and Frank [77] presented a comparison between the continuous and discrete adjoint for quasi-one-dimensional flow. A variation of the discrete field equations proves to be complex for higher order schemes. Due to this limitation of the discrete adjoint approach, early implementations of the discretization of the adjoint equation were consistent only with a first order accurate solution of the flow equation.

Burgreen et al. [10] carried out a second order implementation of the discrete adjoint method for three-dimensional shape optimization of wings for structured grids. Elliot and Peraire [17] used the discrete adjoint method on unstructured meshes for the inverse design of multi-element airfoils and wing-body configurations in transonic flow to produce specified pressure distributions. Anderson and Venkatakrishnan [3] computed optimum shapes for inviscid and viscous flow on unstructured grids using

both the continuous and discrete adjoint methods. Iollo et al. [29] used the continuous adjoint approach to investigate shape optimization on one-and two-dimensional flows. Ta'saan et al. [78] used a one-shot approach with the continuous adjoint formulation. Kim et al. [47] conducted an extensive gradient accuracy study of the Euler and Navier-Stokes equations, concluding that gradients from the continuous adjoint method were in close agreement with those computed by the finite-difference method. A comparison of both the inviscid and viscous continuous and discrete adjoint approaches was conducted by Nadarajah and Jameson [59, 60].

A subject of on-going research is the trade-off between the complexity of the adjoint discretization, the accuracy of the resulting estimate of the gradient, and its impact on the computational cost to approach an optimum solution. An advantage of the continuous adjoint approach is that it provides researchers with an analytical form of the equations which can be studied to understand the nature of the equation and its boundary conditions. Stability analysis and analytical solutions of one-dimensional model problems can be developed to understand the characteristics and behavior of the equation. The scheme used to discretize and march the flow field equations to a steady-state solution can also be employed to solve the discretized continuous adjoint equations. This simplifies the development of codes to implement the method. A disadvantage of the discrete adjoint method is the complexity of applying control theory to the discrete field equations. The complexity of the discretization depends on the sophistication of the flow solver. Reuther [72] stated that for methods that use GMRES (Generalized Minimum Residual) to solve very large linear algebra problems, the task of developing the discrete adjoint equation is as easy as transposing the flux Jacobian matrix. For explicit schemes, such as those used in Jameson's codes, the development of the discrete adjoint equation proves to be a tedious task as shown in Chapter 4. The complete discretization of all the terms in the flow solver requires an extensive amount of algebraic manipulation. The addition of the viscous flux further increases the complexity of deriving its adjoint counterpart. The continuous adjoint method is much simpler to implement for explicit schemes.

Another important issue of interest is the relative accuracy of the gradients derived by the two methods. The continuous adjoint approach provides the inexact

gradient to the exact cost function. On the other hand, the discrete adjoint approach provides the exact gradient to the inexact cost function. Here, the exact cost function is defined as the continuous form of the cost function, and the inexact cost function as the value computed from the discrete field equations and boundary conditions. The continuous gradient is calculated from the discretized continuous adjoint equation, derived from the continuous field equations and cost function. Therefore, the continuous gradient is not necessarily exactly consistent with the cost function which is evaluated numerically. The advantage of the discrete adjoint method is that the resulting discrete gradient is exactly consistent with the discrete cost function. If the discrete gradient is driven to zero, then a local optimum of the discrete cost is attained. However, in the case of the continuous adjoint method, even if its gradient is driven to zero, the discrete objective function may not have converged to the exact discrete minimum. If line searches are used in the optimization algorithm, there may be a conflict, where the discrete minimum in the search direction is inconsistent with the discretized continuous gradient. The discrete adjoint approach does not suffer from this inconsistency. In the limit as the mesh size is reduced, the continuous and discrete adjoint methods should both yield the exact gradient of the continuous cost function.

These questions provide the motivation for a comparison between the continuous and discrete adjoint approaches and methods using gradients obtained by finite-differences or complex-steps. The specific objectives of this work are: first, to review the formulation and development of the viscous adjoint equations for both the continuous and discrete approach; second, to investigate the differences in the implementation of boundary conditions for each method for various cost functions; third, to compare the gradients of the two methods to complex-step gradients for inverse pressure design and drag minimization. Test cases are calculated for various problems and grid sizes.

1.2 Optimum Shape Design for Unsteady Flows

The majority of work in aerodynamic shape optimization has been focused on the design of aerospace vehicles in a steady flow environment. Investigators have applied these advanced design algorithms, particularly the adjoint method, to numerous problems, ranging from the design of two-dimensional airfoils to full aircraft configurations to decrease drag, increase range, and reduce sonic boom as shown in Chapters 4 and 6. These problems have been tackled using many different numerical schemes on both structured and unstructured grids.

Unlike fixed wing aircraft, helicopter rotors and turbomachinery blades operate in unsteady flow and are constantly subjected to unsteady loads. Therefore, optimal control techniques for unsteady flows are needed to improve the performance of helicopter rotors and turbomachinery, and to alleviate the unsteady effects that contribute to flutter, buffeting, poor gust and acoustic response, and dynamic stall.

Helicopter Rotors. The flight envelope of a helicopter rotor is set by the compressibility effects experienced by the advancing rotor blade and the retreating blade dynamic stall. As the helicopter forward flight speed is increased, the freestream velocity observed in the reference frame of the advancing blade is that of the sum of the helicopter forward flight speed and the speed of the advancing blade. At high cruise speeds, the freestream Mach number observed by the advancing blade reaches levels where local supersonic zones on the surface of the rotor blade are present. These regions usually terminate with a shock wave which causes a sudden increase in wave drag. During the retreating phase, the blade incidence approaches the stall angle, causing separation to occur on the upper surface of the blade, which leads to a loss of lift.

At the 38th Cierva Memorial Lecture, Wilby [81] indicated that during the retreating blade stall it is the dramatic change in pitching moment more than the loss of lift that imposes a greater constraint on the design of the rotor blades. The change in pitching moment causes a large oscillatory load on the blade pitch control mechanism which shortens its fatigue life and thus increases the operating cost of the helicopter. Over the years, researchers in the field of rotorcraft aerodynamics have

developed ingenious methods to solve these problems. Their efforts have developed blade profiles that have a high maximum lift coefficient which allows the retreating blade to avoid stall incidence, low wave drag during the advancing phase, and low pitching moment to reduce blade twist and control loads. The introduction of the swept tip to reduce the wave drag and reflex camber towards the rear portion of the upper surface to increase negative loading thus reducing the pitching moment are just some of the technologies introduced over the last twenty years. Unfortunately these improvements only consider the performance of the blade profile for a selected number of flight conditions.

Turbomachinery. The flow through a turbomachine is highly three-dimensional and unsteady due to variation of the hub-to-tip distance along the blade row, blade-to-blade interference, and interactions between the rotors and stators. The nonlinear effects within the turbomachine generally have an unfavorable effect on the flutter characteristics of the blades. Kielb [45] recommended five important questions that must be considered by a designer before venturing into aeroelastic analysis of turbomachineries: first, two-dimensional or three-dimensional flow analysis; second, linear or nonlinear unsteady analysis; third, inviscid or viscous solutions; fourth, single or multi-row setups; and lastly, the type of aero/structural model. He proposes that, for flutter and forced response problems the non-linear unsteady approach is necessary, since linear analysis methods have resulted in significant errors. Apart from aeromechanical issues, turbomachines generate secondary flows, which reduce the efficiency of the system. Vortex shedding and wake-rotor interactions contribute to these losses. These challenges substantiates the need for unsteady design optimization techniques.

A multipoint design approach is one possible technique for the optimization of blade profiles in an unsteady flow environment. This approach only requires a small extension of a steady flow design code in order to redesign a blade or airfoil profile for multiple flow conditions. A typical multipoint design method requires the following three steps: first, steady flow solutions are computed for a number of cases by varying freestream conditions; second, the gradients for each case are computed using either a classical finite-difference method or using an adjoint approach; third, the gradients are weighted and the blade profile is redesigned to satisfy the design objective. Since

the steady flow equations are used to design the blades, uncertainties surrounding the performance of these blades in an unsteady flow environment still prevail.

Recently, the design of blade profiles using unsteady techniques have been attempted. Diverse methods have been employed in the design of rotorcraft and turbomachinery blades. The following are a selected number of papers on this topic. Ghayour and Baysal [23] solved the unsteady transonic small disturbance equation and its continuous adjoint equation to perform an inverse design at Mach 0.6. Aerodynamic shape optimization of rotor airfoils in an unsteady viscous flow was attempted by Yee et al. [82] using a response surface methodology. Here the authors used an upwind-biased-factorized implicit numerical scheme to solve the RANS equations with a Baldwin-Lomax turbulence model. A response surface methodology was then employed to optimize the rotor blade. The objective function was a sum of the L/D at three different azimuth angles and was later redefined to include unsteady aerodynamic effects. Florea and Hall [20] modeled a cascade of turbomachinery blades using the steady and time-linearized Euler equations. Gradients for aeroelastic and aeroacoustic objective functions were then computed using the discrete adjoint approach. Both the flow and adjoint equations were solved using a finite-volume Lax-Wendroff scheme. The gradients were then used to improve the aeroelastic stability and acoustic response of the airfoil.

In this work, a framework is developed to perform sensitivity analysis for unsteady transonic flow, and to use this information to modify the shape of the profile in a favorable manner. Optimal control of time dependent trajectories is generally complicated by the need to solve the adjoint equation in reverse time from a final boundary condition. This requires information from the trajectory solution, which in turn depends on the control derived from the adjoint solution. The method presented in this dissertation is restricted to periodic unsteady transonic flows. The time accurate adjoint equations are based on Jameson's cell-centered multigrid-driven fully-implicit scheme with upwind-biased blended first and third order artificial dissipation fluxes [36].

In this dissertation both the time accurate continuous and discrete adjoint equations are derived, and then used in the redesign of the RAE 2822 and VR-7 rotor airfoils undergoing a pitching oscillation to achieve lower time-averaged drag, while

keeping the time-averaged lift constant [61]. The fully unsteady design technique is compared to multipoint and steady adjoint approaches to gauge its effectiveness.

1.3 The Remote Inverse Design Problem

A 2001 National Research Council study [67] concluded that, "...the sonic boom is the major barrier to the development of the supersonic business jet and a major, but not the only, barrier to the development of supersonic transports with overland capability." The committee also determined that there is a potential market for at least 200 supersonic business jets over a ten year period. The 8-15 passenger jets will probably fly at approximately Mach 1.8 with a range of $3000 - 4000$ nautical miles.

Currently, the only commercial supersonic aircraft in operation is the Concorde, built jointly by France and Britain. Twenty aircraft have been built since 1973 and currently only twelve are in operation. The aircraft was built to cruise at Mach 2.0 with a total range of 4090 nautical miles at 60,000 feet. Designs of supersonic transports of the future will benefit from multidisciplinary optimization techniques that were not available during the design and construction of the Concorde. Apart from low sonic boom capability, the new breed of supersonic transports must also possess superior performance characteristics compared to its predecessors to compete in the modern commercial jet industry. These include improvements in structures, aerodynamics, and propulsion. In particular, the experience of NASA's HSR program suggests that it should be possible to improve the lift-to-drag ratio from the value of 7.5 attained by the Concorde to around 9 [12].

Before the sonic boom reduction problem can be attempted, it is important to have the capability to calculate the sonic boom or ground pressure signature accurately. For typical cruise altitudes required for aircraft efficiency, the distance from the source of the acoustic disturbance to the ground is typically greater than 50,000 ft. A reasonably accurate propagation of the pressure signature can only be obtained with small computational mesh spacings that would render the analysis of the problem intractable for even the largest parallel computers. An approach that has been

used successfully in the past is the use of near to far field extrapolation of pressure signatures based on principles of geometrical acoustics and non-linear wave propagation. These methods are based on the solutions of simple ordinary differential equations for the propagation of the near field pressure signature to the ground. The Whitham F-function [80] and Thomas equivalent waveform parameter method [79] are common methods of choice.

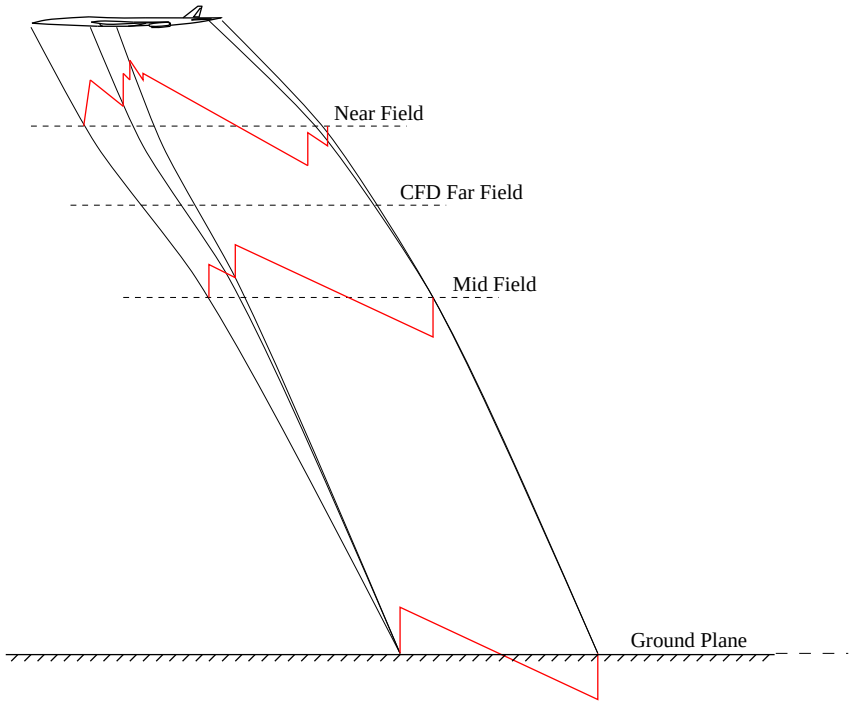


Figure 1.1: Schematic of the Propagation of the Aircraft Pressure Signature

Figure 1.1 is a schematic of the propagation of the aircraft pressure signature. 'CFD Far Field' indicates the far field boundary of the mesh. At a pre-specified distance below the aircraft and still within the CFD mesh, the location of a near field plane can be seen. This plane is the effective interface between the CFD solution and the wave propagation program. At the near field plane, the flow solution w_o is represented using a number of parameters, M, which can be taken as the number of mesh points on which the pressure waveform has a value significantly different from the free stream. The lower portion of the domain between the CFD near field

and the ground plane is where the pressure signature propagation method will be active. Given the conditions, w_o, the propagation altitude, and the altitude dependent atmospheric properties $\rho(z), p(z), T(z)$, the propagation method produces a pressure signature at the ground plane we are interested in, which can be used to determine any of a variety of measures of sonic boom impact such as overpressures, rise time, impulse, etc.

Traditional methods to reduce the sonic boom signature were targeted towards reducing aircraft weight, increasing lift-to-drag ratio, improving the specific fuel consumption, etc. Seebass and Argrow [76] revisited sonic boom minimization and provided a detailed study of sonic boom theory and figures of merit for the level of sonic booms. Through the support of the DARPA QSP Program, advanced algorithms for the design and optimization of quiet supersonic platforms have been developed at Stanford during the last two years. DARPA's vision for the project is to develop conceptual aircraft designs that produce initial overpressures of 0.3psf, while cruising at Mach 2.5 with a range of 6000 nautical miles and with a weight of 100,000lbs. This is an ambitious reduction in the initial overpressure compared to the Concorde's $1.5 - 2.0$psf. Our experience has indicated that large reductions in the ground peak pressure cannot be achieved with minor shape modifications of the baseline configuration. Alternative design methods such as genetic algorithms have been used in a multi-level design environment to get in the neighborhood of the optimum design before switching over to a gradient-based method to refine the design. Promising results have been achieved by using genetic algorithms in a linear method environment. Nonlinear methods are needed in order to meet several goals: first, to verify if not improve the results of the linear based method; second, to improve the design by exploring the techniques of optimal control; lastly, to allow the introduction of more objective functions to improve the final design.

The concept presented in this work proposes the idea that the ground pressure signature could be adjusted by modifying the aircraft surface geometry to control the near field pressure distribution. It is not at all clear what type of changes to the surface geometry would produce near field pressure distributions whose propagation to the ground would generate sonic booms with lower peaks. It appears, however, that

the problem might be separated into two parts: first, the identification of near field pressure distributions that are both feasible and lead to acceptable ground signatures; and second, the design of the surface geometry such that it will produce the desired near field pressure distribution.

Traditional adjoint implementations were aimed at reducing a cost function computed from the pressure distribution on the surface that is being modified. In this case, however, we would like to obtain the sensitivity of pressure distributions at points remotely located from the surface where the geometry is being modified. This type of sensitivity calculation has not been attempted before in aerodynamic shape design, but is closely related to inverse scattering problems in acoustics and electromagnetics. In such an approach, a target near field pressure distribution must be specified. The cost function may then be chosen as the integral of the square of the difference between the current and target near field pressure distribution. The gradient of the cost function with respect to the design variables such as the surface mesh points is calculated, and a direction of improvement is obtained from an optimization algorithm. The procedure is repeated until a new aircraft surface geometry is produced that provides a near field pressure signature that approaches the specified target near field pressure distribution, provided that it is actually realizable. The design procedure should also include other objective functions and constraints to maintain or improve other aircraft performance parameters such as lift-to-drag ratio. The possibility that the adjoint method could be adapted to solve the remote inverse problem was demonstrated by Nadarajah and Jameson [62] for a two-dimensional internal flow problem. The method was then extended for three-dimensional wing and wing-body configurations in supersonic flow by Nadarajah et al. [63, 64, 65].

The issue of choosing a near field signature to produce a desired ground signature was addressed by Alonso et al. [2]. The work accomplished in this research focuses on controlling the near field signature, and not the ground signature. A future extension of the method would be to include the wave propagation program into the design procedure, such that the ground pressure signature is considered as the target pressure distribution instead of the near field pressure distribution.

Chapter 2

The Euler and Navier-Stokes Equations

This chapter presents the mathematical models used in this work. Section 2.1 describes the integral and conservative forms of the field equations and its boundary conditions. Section 2.2 presents the numerical discretization of the mathematical models, artificial dissipation schemes, and lastly convergence acceleration procedures such as local time stepping, residual averaging, and multigrid.

2.1 Mathematical Model

The following sub-sections describe the development of the integral and conservative forms of the field equations and their boundary conditions.

2.1.1 Conservation of Mass

Let $V(t)$ be a control volume with a moving surface $S(t)$ and an outward pointing normal vector $\vec{n}$. The statement "conservation of mass" implies that mass is neither created nor destroyed in the control volume. For the case of no production or annihilation in V, then the time rate of change of mass in V must equal zero and the

statement can be expressed mathematically as

$$\frac{d}{dt}\int_{V(t)} \rho \, dV = 0.$$

By the *Reynolds' Transport theorem* the above equation can be rewritten as

$$\frac{d}{dt}\int_{V(t)} \rho \, dV = \int_{V(t)} \left[\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{u})\right] \, dV = 0. \tag{2.1}$$

The divergence of ρu can be expanded to $\vec{u} \cdot \nabla \rho + \rho \nabla \cdot \vec{u}$. Using the definition of a material derivative, the material derivative of ρ can be defined as

$$\frac{D\rho}{Dt} = \frac{\partial \rho}{\partial t} + \vec{u} \cdot \nabla \rho.$$

The statement for conservation of mass from equation (2.1) can be written in material derivative form as

$$\frac{d}{dt}\int_{V(t)} \rho \, dV = \int_{V(t)} \left[\frac{D\rho}{Dt} + \rho \nabla \cdot \vec{u}\right] \, dV = 0.$$

2.1.2 Conservation of Momentum

Let F_i be the total force acting on all fluid particles in the volume $V(t)$ along the x_i direction. Then by Newton's second law, the time rate of change of momentum is equal to the sum of the surface and body forces acting on all fluid particles. The time rate of change of momentum in the x_i direction can be expressed as

$$F_i = \frac{d}{dt}\int_{V(t)} \rho u_i \, dV.$$

Define $\vec{G_i}$ to be the body force per unit volume vector applied to all fluid particles in volume $V(t)$ and $\vec{T_i}$ as the surface force per unit area stress vector acting on the control surface $S(t)$. Then the sum of the surface and body forces can be formulated as

$$F_i = \int_{S(t)} \vec{T_i} \, dS + \int_{V(t)} \rho \vec{G_i} \, dV.$$

By the application of *Reynolds' Transport theorem*, the time rate of change of momentum can be written in material derivative form and thus the integral form of the conservation of momentum can be defined as

$$\int_{V(t)} \rho \frac{Du_i}{Dt}\, dV = \int_{S(t)} \vec{T_i}\, dS + \int_{V(t)} \rho \vec{G_i}\, dV. \tag{2.2}$$

Equation (2.2) consists of two volume integrals and one surface integral. The surface integral can be reformulated into a volume integral and equation (2.2) can be expressed as a single volume integral. First, define the vector $\vec{T_i}$ as the sum of the hydrostatic pressure and viscous stress

$$\vec{T_i} = \sigma_{ki} n_k, \tag{2.3}$$

where σ_{ki} is the sum of the viscous stress tensor and pressure as stated below:

$$\begin{aligned} \sigma_{ki} &= -p\delta_{ki} + \tau_{ki}, \\ p &= \text{Pressure.} \\ \delta_{ki} &= \text{Kronecker delta function.} \\ \tau_{ki} &= \text{Viscous stress tensor.} \end{aligned}$$

Then the integral of the surface force $\vec{T_i}$ over the control surface $S(t)$ can be expressed as

$$\begin{aligned} \int_{S(t)} \vec{T_i}\, dS &= \int_{S(t)} \sigma_{ki} n_k\, dS \\ &= \int_{S(t)} \left(-p\delta_{ki} + \tau_{ki}\right) n_k\, dS \\ &= \int_{S(t)} -p\delta_{ki} n_k + \tau_{ki} n_k\, dS. \end{aligned}$$

By application of the *Gauss' theorem*, the surface integral is converted to a volume integral and we have

$$\begin{aligned}\int_{S(t)} \vec{T}_i \, dS &= \int_{S(t)} -p\delta_{ki}n_k + \tau_{ki}n_k \, dS \\ &= \int_{V(t)} -\frac{\partial}{\partial x_k}\left(p\delta_{ki}\right) + \frac{\partial \tau_{ki}}{\partial x_k} \, dV \\ &= \int_{V(t)} -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ki}}{\partial x_k} \, dV. \end{aligned} \tag{2.4}$$

Substitute equation (2.4) into equation (2.2) to yield the integral form of the conservation of momentum

$$\int_{V(t)} \left[\rho \frac{Du_i}{Dt} + \frac{\partial p}{\partial x_i} - \frac{\partial \tau_{ki}}{\partial x_k} - \rho \vec{G}_i\right] \, dV = 0.$$

If we assume that the fluid has the property that the elements of the viscous stress tensor are related linearly with the elements of the rate of deformation tensor (Newtonian Fluid) and the gas is isotropic, then the viscous stress tensor can be defined as

$$\tau_{ki} = \mu\left[\frac{\partial u_k}{\partial x_i} + \frac{\partial u_i}{\partial x_k}\right] + \lambda\left[\frac{\partial u_j}{\partial x_j}\right]\delta_{ki}. \tag{2.5}$$

Based on Stokes assumption, $\lambda = -2\mu/3$.

2.1.3 Conservation of Energy

The rate of change of energy in volume $V(t)$ is affected by four factors: first, the rate of work done by the surface force per unit area stress vector; second, the rate of work done by the body force per unit volume vector; third, the rate of heat loss through the control surface $S(t)$; and last, the rate of volumetric heat addition. The conservation of energy equation can then be defined as

$$\frac{d}{dt}\int_{V(t)} \rho\left(e + \frac{u^2}{2}\right) \, dV = \int_{S(t)} \vec{T}_k u_k \, dS + \int_{V(t)} \rho \vec{G}_k u_k \, dV - \int_{S(t)} q_k n_k \, dS + \int_{V(t)} Q \, dV.$$

The left-hand side of the above equation could be transformed into a material derivative form using the *Reynolds' Transport theorem* and the surface integrals could be converted to volume integrals with the *Gauss' theorem*. Substitution of equation (2.3) into the above equation will produce

$$\int_{V(t)} \left[\rho \frac{D}{Dt} \left(e + \frac{u^2}{2} \right) + \frac{\partial}{\partial x_k} \left(-\sigma_{ki} u_i + q_k \right) - \rho \vec{G}_k u_k - Q \right] \, dV = 0,$$

where e is the internal energy and q_k is defined by the Fourier law of heat conduction:

$$q_k = -\kappa \frac{\partial T}{\partial x_k}, \quad \kappa = \text{ coefficient of thermal conductivity.} \tag{2.6}$$

2.1.4 Conservative Form of the Field Equations

In order to allow for geometric shape changes it is convenient to use a body fitted coordinate system, so that the computational domain is fixed. This requires the formulation of the Navier-Stokes equations in a transformed coordinate system. The Cartesian coordinates and velocity components are denoted by x_1, x_2, x_3 and u_1, u_2, u_3. Einstein notation simplifies the presentation of the equations, where summation over $k = 1$ to 3 is implied by a repeated index k. The three-dimensional Navier-Stokes equations then take the conservative form,

$$\frac{\partial w}{\partial t} + \frac{\partial f_i}{\partial x_i} = \frac{\partial f_{v_i}}{\partial x_i} \quad \text{in } V, \tag{2.7}$$

where the state vector w, inviscid flux vector f, and viscous flux vector f_v are described respectively by

$$w = \begin{Bmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ \rho E \end{Bmatrix}, \quad f_i = \begin{Bmatrix} \rho u_i \\ \rho u_i u_1 + p\delta_{i1} \\ \rho u_i u_2 + p\delta_{i2} \\ \rho u_i u_3 + p\delta_{i3} \\ \rho u_i H \end{Bmatrix}, \quad \text{and } f_{vi} = \begin{Bmatrix} 0 \\ \tau_{ij}\delta_{j1} \\ \tau_{ij}\delta_{j2} \\ \tau_{ij}\delta_{j3} \\ u_j \tau_{ij} + k\frac{\partial T}{\partial x_i} \end{Bmatrix}. \tag{2.8}$$

In these definitions, ρ is the density, E is the total energy and δ_{ij} is the Kronecker delta function. The pressure is determined by the equation of state

$$p = (\gamma - 1)\,\rho \left[E - \frac{1}{2}\,(u_i u_i) \right], \tag{2.9}$$

where γ is the ratio of the specific heats and the stagnation enthalpy is given by

$$H = E + \frac{p}{\rho}.$$

The viscous stresses may be written as

$$\tau_{ij} = \mu \left[\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right] + \lambda \left[\frac{\partial u_k}{\partial x_k} \right] \delta_{ij}, \tag{2.10}$$

where μ and $\lambda = -2\mu/3$ are the first and second coefficients of viscosity. The viscosity coefficient relates the momentum flux to the velocity gradient. Assuming *Newtonian fluid*, the applied shear varies linearly with the strain rate. The constant of proportionality is the coefficient of viscosity, which is a thermodynamic property that varies with pressure and temperature. A common approximation for the coefficient of viscosity is the Sutherland equation defined as

$$\mu = C_1 \frac{T^{\frac{3}{2}}}{T + C_2}. \tag{2.11}$$

where C_1 and C_2 are constants for a given gas. For air at moderate temperatures, $C_1 = 1.458\text{x}10^{-6} kg/(ms\sqrt{^oK})$ and $C_2 = 110.4^oK$. The coefficient of thermal conductivity relates the heat flux to the temperature gradient. The coefficient and the temperature are computed as

$$k = \frac{c_p \mu}{Pr}, \qquad T = \frac{p}{R\rho},$$

where Pr is the Prandtl number, c_p is the specific heat at constant pressure, and R is the gas constant.

2.1.5 Boundary Conditions

We now apply boundary conditions to complete the mathematical model. At the solid wall the boundary conditions for inviscid and viscous flows can be defined as follows. For inviscid flow, the velocity at the wall must be tangent to the slope of the wall. This corresponds to a zero flux through the wall and thus

$$(\vec{u} \cdot \vec{n})_{wall} = 0,$$

where $\vec{n}$ is the wall surface unit normal vector. For viscous problems, the no-injection and no-slip conditions are imposed and requires an additional boundary condition to the one above

$$\left(\vec{u} \cdot \vec{t}\right)_{wall} = 0,$$

where $\vec{t}$ is the wall unit tangent vector. This effectively means that the velocity at the wall is zero. The above boundary conditions satisfy the momentum equation. For the case of the energy equation, either an adiabatic or isothermal boundary condition is required. In this work, an adiabatic boundary condition is employed and defined as

$$(\vec{q} \cdot \vec{n})_{wall} = 0.$$

This translates to a zero heat flux through the normal of the wall.

The solution is also required to exhibit the correct asymptotic behavior in the far-field. When a finite computational domain is used one should distinguish inflow and outflow boundaries. At any boundary, only conditions corresponding to incoming waves should be specified. Moreover it is beneficial to use procedures that reduce the reflection of outgoing waves, as will be discussed in section 2.2.3.

2.1.6 Compressible Reynolds Averaged Navier-Stokes Equations

The Navier-Stokes equations are generally accepted by the scientific community to describe the behavior of fluid flow. However, at high Reynolds numbers the onset of

turbulence possess a challenge to accurately solve the equations. A common method to solve the equation at such conditions is to statistically average the variables. A generic variable m is substituted by the mean value, $\overline{m}$, and the instantaneous value, m''. Such treatment of the variables allows local fluctuations of these variables in the flow and this forms the basis for turbulent flows. The mean value of the variable is the time-averaged value as described by the following equation

$$\overline{m} = \int_{t_o}^{t_o+T} m \, dt,$$

where the variables t_o marks the start of the flow and T is the time interval. The time interval T is assumed to be longer than the characteristic time scale of the turbulence. This is to allow the turbulent aspect of the flow to be fully developed and thus the mean value of the flow variable would better represent the mean behavior of the flow over a period of time. Next, the mass weighted average of the generic variable m is computed using the following equation

$$\widetilde{m} = \frac{\overline{\rho m}}{\rho}.$$

The local instantaneous fluctuation is obtained by the subtraction of the mean value of the variable from its instantaneous value such as $m'' = m - \widetilde{m}$. Next, substitute the velocity component, pressure, and internal energy with the mean and instantaneous values. Lastly, take the time-average of the equations to yield the Reynold Averaged Navier-Stokes equations

$$\frac{\partial \widehat{w}}{\partial t} + \frac{\partial \widehat{f_i}}{\partial x_i} = \frac{\partial \widehat{f_{v_i}}}{\partial x_i} \quad \text{in } V, \tag{2.12}$$

where the state vector $\widehat{w}$, inviscid flux vector $\widehat{f}$, and viscous flux vector $\widehat{f}_v$ are described respectively by

$$\widehat{w} = \begin{Bmatrix} \overline{\rho} \\ \overline{\rho}\widetilde{u}_1 \\ \overline{\rho}\widetilde{u}_2 \\ \overline{\rho}\widetilde{u}_3 \\ \overline{\rho}\widetilde{E} \end{Bmatrix}, \quad f_i = \begin{Bmatrix} \overline{\rho}\widetilde{u}_i \\ \overline{\rho}\widetilde{u}_i\widetilde{u}_1 + \widetilde{p}\delta_{i1} \\ \overline{\rho}\widetilde{u}_i\widetilde{u}_2 + \widetilde{p}\delta_{i2} \\ \overline{\rho}\widetilde{u}_i\widetilde{u}_3 + \widetilde{p}\delta_{i3} \\ \overline{\rho}\widetilde{u}_i\widetilde{E} + \widetilde{u}_i\overline{p}\delta_{ij} \end{Bmatrix}, \quad \text{and}$$

$$f_{vi} = \begin{Bmatrix} 0 \\ \left(\overline{\tau}_{ij} - \overline{\rho u_1'' u_i''}\right)\delta_{j1} \\ \left(\overline{\tau}_{ij} - \overline{\rho u_2'' u_i''}\right)\delta_{j2} \\ \left(\overline{\tau}_{ij} - \overline{\rho u_3'' u_i''}\right)\delta_{j3} \\ \widetilde{u}_j\left(\overline{\tau}_{ij} - \overline{\rho u_i'' u_j''}\right) - \overline{u_i''\left(-p\delta_{ij} + \tau_{ij} - \frac{1}{2}\rho u_i'' u_j''\right)} + \overline{q}_j + \overline{\rho u_j'' e''} \end{Bmatrix}. \quad (2.13)$$

Note here that deletion of the instantaneous fluctuation terms marked by the ($''$) sign reduces the above equations to the original familiar Navier-Stokes form. The second term in the energy dissipation flux term in equation (2.13) represents the mean energy dissipation and its magnitude is generally several orders of magnitude smaller than the other terms and thus often neglected. Now it is clearly visible that the majority of the difference is in the addition of extra terms to the viscous stress and the heat flux. They can be redefined as

$$\begin{aligned} \tau_{ij_{total}} &= \overline{\tau}_{ij} - \overline{\rho u_i'' u_j''} \\ q_{j_{total}} &= \overline{q}_j + \overline{\rho u_j'' e''}. \end{aligned}$$

The next procedure would be to define and calculate the additional terms. Since it is not possible to solve for these terms directly, they are often calculated by introducing additional equations that relate the higher order to the lower order correlation terms.

The various ways these quantities can be represented and calculated are part of turbulence modeling. This area of research is beyond the scope of the present purpose and will not be discussed here. In 1877, Boussinesq [7] proposed a scalar that relates the apparent turbulent shearing stress with the rate of mean strain. The result is an additional coefficient of viscosity term deemed the eddy viscosity that is added to the original coefficient of viscosity that represents the laminar aspects of the flow. Additional algebraic models are than introduced to calculate the eddy viscosity. The new total stress tensor $\tau_{ij_{total}}$ can then be defined as

$$\tau_{ij} = (\mu_{lam} + \mu_{turb}) \left\{ \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \left[\frac{\partial u_k}{\partial x_k} \right] \right\} \delta_{ij},$$

where μ_{lam} is the laminar coefficient of viscosity defined by equation (2.11) and μ_{turb} is the eddy viscosity coefficient. The next sub-section describes a simple algebraic model to calculate the eddy viscosity.

2.1.7 Baldwin-Lomax Turbulence Model

The Baldwin and Lomax turbulence model [4] is an algebraic model for the determination of the eddy viscosity. The dual layered eddy viscosity formulation is sufficient to complete the Reynolds Averaged Navier-Stokes equation. The turbulent eddy viscosity coefficient can be calculated as

$$\frac{\mu_{turb}}{\mu_\infty} = \begin{cases} (\mu^*_{turb})_{inner}, & \text{when } y^* < y^*_{crossover} \\ (\mu^*_{turb})_{outer}, & \text{when } y^*_{crossover} < y^* \end{cases}, \tag{2.14}$$

where $y^*_{crossover}$ is the minimum value of the dimensionless normal distance to the wall, y^*, at which the inner and outer eddy viscosity formulations produce the same result. The inner eddy viscosity coefficient follows the formula introduced by Prandtl-Van Driest and is defined as

$$(\mu^*_{turb})_{inner} = \frac{Re_\infty}{M_\infty \sqrt{\gamma}} \rho^* {l^*}^2 |\omega^*|,$$

where l^* is the length scale of the turbulence in the inner region and ω^* is the magnitude of the dimensionless vorticity vector. The length scale can be further defined as

$$l^* = k\frac{y}{c}\left(1 - e^{-\frac{y^+}{A^+}}\right),$$

where k is a constant with a typical value of 0.4 and y^+ is the dimensionless distance from the wall and defined as

$$y^+ = \frac{\sqrt{\rho^*_{wall}\tau^*_{wall}y^*}}{\mu^*_{wall}}\sqrt{\frac{Re_\infty}{M_\infty\sqrt{\gamma}}}.$$

Next we define the eddy viscosity for the outer boundary layer as

$$(\mu^*_{turb})_{outer} = \frac{Re_\infty}{M_\infty\sqrt{\gamma}}KC_{cp}\rho^*F_{wake}F_{kleb},$$

where K and C_{cp} are constants with typical values of 0.0168 and 1.6, and F_{wake} and F_{kleb} are defined using the following formulas. First, let us define the function F as

$$F = y^*|\omega^*|\left(1 - e^{-\frac{y^{+*}}{A^*}}\right).$$

Then

$$F_{wake} = \min\left(y^*_{max}F_{max}, C_{wk}\frac{y^*_{max}U^{*2}_{max}}{F_{max}}\right),$$

where C_{wk} is a constant with a typical value of 1.0, U_{max} is the maximum difference of the magnitude of the velocity profile, and y_{max} is the location that maximizes the function F as defined above. Then the value F_{max} is the corresponding maximum value of the function. The value for F_{kleb} is a correction that accounts for intermittency effects as defined as

$$F_{kleb} = \left\{1 + 5.5\left(\frac{C_{kleb}y}{y_{max}}\right)^6\right\}^{-1}.$$

In order to monitor the transition from laminar to turbulence using the Baldwin-Lomax algebraic turbulence model a transition coefficient is used to turn on or off the contribution from the model. Usually the eddy viscosity is set to zero if $\mu^*_{max} <$

$C_{transition}$. A typical value for $C_{transition}$ is usually 14.

2.2 Numerical Discretization

In order to carry out a computational simulation of a complex nonlinear partial differential equation, a numerical discretization procedure is needed to transform it into a set of algebraic equations. The first step is to represent the continuous domain by a mesh of discrete points, where the dependent variables of the partial differential equation are represented. Mesh generation has become an important field of study to enable solutions for more complex geometries. The choice of the type of mesh is usually based on the complexity of the geometry and the desired level of accuracy and approximation of the continuous problem. In this work, several different types of mesh have been used to achieve the desired results. In Chapter 4 where the Navier-Stokes equations are used in the design of airfoil shapes in a two-dimensional viscous flow environment, a high mesh resolution close to the surface of the airfoil is required to resolve the boundary layer and its interaction with the shock wave. However, in Chapters 5 and 6, where only the Euler equations are employed, large gradients close to the surface of the airfoil except for the shock wave do not exist and more uniform and regular meshes are sufficient to provide accurate numerical approximations.

The next task at hand is to decide on the choice of a numerical scheme between the finite-difference, finite-volume, and finite-element methods. The ultimate question is whether the numerical solution is a good approximation of the exact solution. The stability and robustness of the scheme are also important factors. The next subsections describe the numerical discretization used in this work. The discussion is restricted to the case of two-dimensional flow. Similar methods have been used in the three-dimensional calculations presented in Chapter 6. The various convergence acceleration techniques used such as local time stepping, residual averaging, and multigrid are presented.

2.2.1 Finite-Volume Technique

A finite-volume methodology is used to discretize the integral form of the conservation laws. The method is preferred over the traditional finite-difference formulation because it allows an arbitrary mesh. It also has the advantage that it preserves the conservation laws at a discrete level, ensuring global conservation of mass, momentum, and energy.

When using a discretization on a body conforming structured mesh, it is useful to consider a transformation to the computational coordinates (ξ_1,ξ_2) defined by the metrics

$$K_{ij} = \left[\frac{\partial x_i}{\partial \xi_j}\right], \quad J = \det(K), \quad K_{ij}^{-1} = \left[\frac{\partial \xi_i}{\partial x_j}\right].$$

Figure 2.1 illustrates the finite-volume mesh for cell (i, j). The values of the flow properties are stored at the cell centers marked by the red dot. Points between cell centers are marked as black dots and appear along cell boundaries and are identified by the $\pm\frac{1}{2}$ to either i or j. The cell boundaries will also be referred to as flux faces.

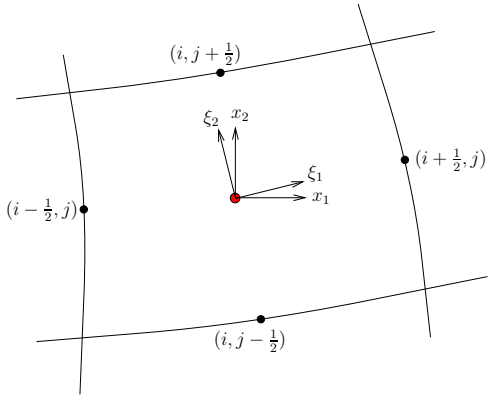


Figure 2.1: Finite-Volume Mesh for Cell (i, j)

The Navier-Stokes equations can then be written in computational space as

$$\frac{\partial (Jw)}{\partial t} + \frac{\partial (F_i - F_{v_i})}{\partial \xi_i} = 0 \quad \text{in } \mathcal{D}, \tag{2.15}$$

where the inviscid and viscous flux contributions are now defined with respect to the computational cell faces by $F_i = S_{ij} f_j$ and $F_{v_i} = S_{ij} f_{v_j}$, and the quantity $S_{ij} = JK_{ij}^{-1}$ represents the projection of the ξ_i cell face along the x_j axis. In obtaining equation (2.15) we have made use of the property that

$$\frac{\partial S_{ij}}{\partial \xi_i} = 0,$$

which represents the fact that the sum of the face areas over a closed volume is zero, as can be readily verified by a direct examination of the metric terms. When equation (2.15) is formulated for each computational cell, a system of first order ordinary differential equations is obtained. Equation (2.15) can then be written for each computational cell as

$$\frac{\partial (Jw)_{ij}}{\partial t} + R(w)_{i,j} = 0, \tag{2.16}$$

where

$$R(w)_{ij} = \frac{\partial F_1}{\partial \xi_1} + \frac{\partial F_2}{\partial \xi_2} - \frac{\partial F_{v_1}}{\partial \xi_1} - \frac{\partial F_{v_2}}{\partial \xi_2}. \tag{2.17}$$

Then each partial derivative in equation (2.17) represents the net flux in each direction of the computational space. Each partial derivative of the convective and viscous flux gradients can then be represented in discrete form for each computational cell using a central second order discretization as

$$\frac{\partial F_1}{\partial \xi_1} = f_{i+\frac{1}{2},j} - f_{i-\frac{1}{2},j}, \tag{2.18}$$

where the $\pm\frac{1}{2}$ notation indicates that the quantity is calculated at the flux faces. Expanding each partial derivative to its discrete form in equation (2.17) using the notation from equation (2.18) results to

$$\begin{aligned} R(w)_{ij} = \quad & f_{i+\frac{1}{2},j} - f_{i-\frac{1}{2},j} + f_{i,j+\frac{1}{2}} - f_{i,j-\frac{1}{2}} \\ - \quad & f_{v_{i+\frac{1}{2},j}} + f_{v_{i-\frac{1}{2},j}} - f_{v_{i,j+\frac{1}{2}}} + f_{v_{i,j-\frac{1}{2}}}, \end{aligned} \tag{2.19}$$

where f and f_v are the convective and viscous fluxes.

As it stands, this scheme would reduce to a second order accurate central difference scheme if the mesh was Cartesian. Schemes of this type generate oscillations around shock discontinuities and allow odd-even decoupling of the solution. To damp these high frequency errors, artificial dissipation terms are added to the convective and viscous fluxes. A detailed discussion of the need for artificial dissipation and the scheme used in this work is presented in sub-section 2.2.2. Equation (2.19) can then be rearranged to produce

$$R(w)_{ij} = h_{i+\frac{1}{2},j} - h_{i-\frac{1}{2},j} + h_{i,j+\frac{1}{2}} - h_{i,j-\frac{1}{2}}, \tag{2.20}$$

where h denotes the numerical flux across the cell interfaces,

$$h_{i+\frac{1}{2},j} = f_{i+\frac{1}{2},j} - f_{v_{i+\frac{1}{2},j}} - d_{i+\frac{1}{2},j}, \tag{2.21}$$

and d represents the artificial dissipation term.

Discretization of the Convective Flux

In the finite-volume technique used here, the values of the flow variables are stored at the cell centers, and can be regarded as cell averages. Accordingly the convective flux $f_{i+\frac{1}{2},j}$ at the cell face as shown in figure 2.2, is computed by taking the average of the flux contributions from each cell across the cell face as shown in the following equation

$$f_{i+\frac{1}{2},j} = \frac{1}{2}\left(f^{+}_{i+1,j} + f^{-}_{i,j}\right). \tag{2.22}$$

On a Cartesian grid, the discretization produces a three-point second order central difference scheme for each flux gradient. Next we define the flux velocity in each cell as

$$\begin{aligned} q^{-} &= \frac{y_{\eta_{i+\frac{1}{2},j}}(\rho u)_{i,j} - x_{\eta_{i+\frac{1}{2},j}}(\rho v)_{i,j}}{\rho_{i,j}} \\ q^{+} &= \frac{y_{\eta_{i+\frac{1}{2},j}}(\rho u)_{i+1,j} - x_{\eta_{i+\frac{1}{2},j}}(\rho v)_{i+1,j}}{\rho_{i+1,j}}. \end{aligned}$$

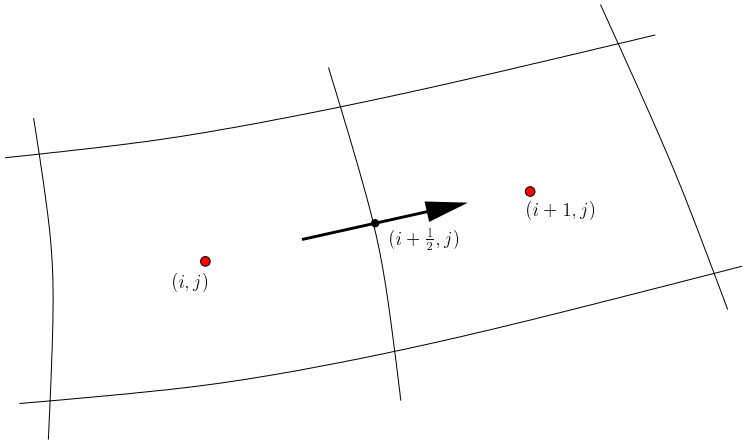


Figure 2.2: Discretization of the Convective Fluxes

Then the flux vectors can be formulated as

$$f^+_{i+1,j} = \left\{ \begin{array}{c} \rho_{i+1,j} q^+ \\ (\rho u)_{i+1,j}\, q^+ + y_{\eta_{i+\frac{1}{2},j}} p_{i+1,j} \\ (\rho v)_{i+1,j}\, q^+ - x_{\eta_{i+\frac{1}{2},j}} p_{i+1,j} \\ (\rho E + p)_{i+1,j}\, q^+ \end{array} \right\} \text{,and } \; f^-_{i,j} = \left\{ \begin{array}{c} \rho_{i,j} q^- \\ (\rho u)_{i,j}\, q^- + y_{\eta_{i+\frac{1}{2},j}} p_{i,j} \\ (\rho v)_{i,j}\, q^- - x_{\eta_{i+\frac{1}{2},j}} p_{i,j} \\ (\rho E + p)_{i,j}\, q^- \end{array} \right\} \quad (2.23)$$

Discretization of the Viscous Flux

A numerical evaluation of the viscous flux requires an estimate of the partial derivative of velocity in the viscous stress tensor from equation (2.5) and the partial derivative of temperature from the Fourier Law of heat conduction in equation (2.6). To evaluate the viscous flux at the cell face, we first compute the stress tensor and the heat flux components of the viscous flux at the end points (vertex) of the edge by employing a discrete Gauss theorem to the auxiliary control volume formed by the cell centers of the four cells containing the vertex $(i+\frac{1}{2}, j \pm \frac{1}{2})$ as illustrated in figure 2.3. Second, the viscous flux across the cell face is computed by averaging the viscous fluxes at both ends of the edge.

The viscous flux $f_{v_{i+\frac{1}{2},j}}$ at the cell face can be computed by taking the average of

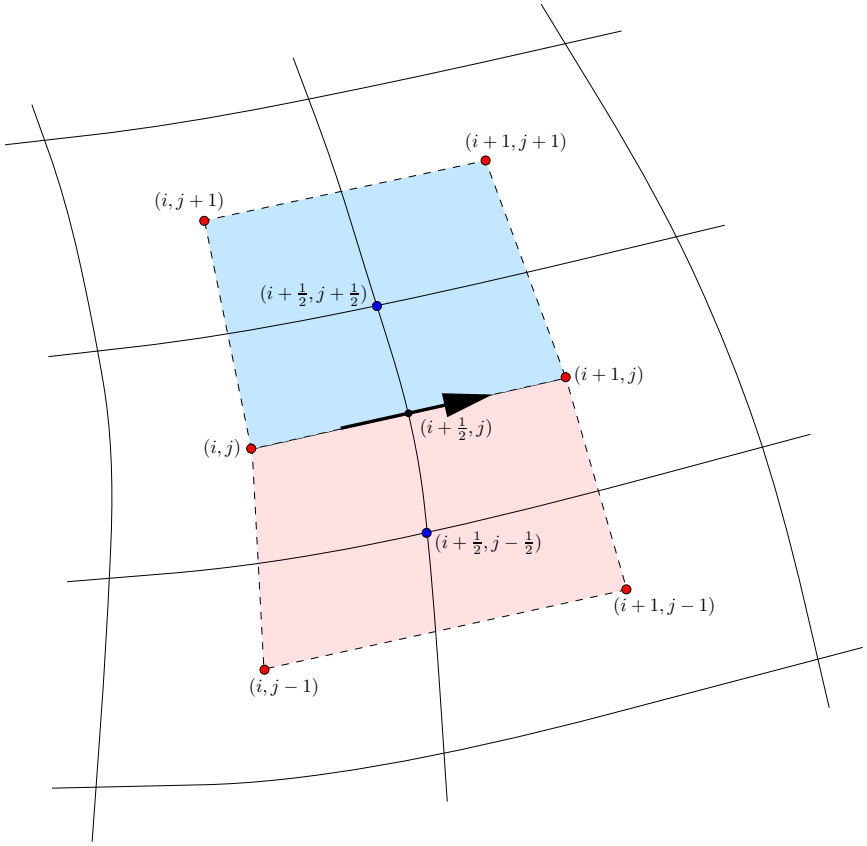


Figure 2.3: Auxiliary Control Volume for the Discretization of the Viscous Flux

the fluxes computed at the cell vertex as shown below

$$f_{v_{i+\frac{1}{2},j}} = y_{\eta_{i+\frac{1}{2},j}} f^*_{v_{i+\frac{1}{2},j}} + x_{\eta_{i+\frac{1}{2},j}} g^*_{v_{i+\frac{1}{2},j}}, \tag{2.24}$$

where f^* and g^* represent the fluxes at the mid-point of the cell face. The fluxes are computed by taking the average of the flux contributions from the cell vertex at $(i+\frac{1}{2}, j \pm \frac{1}{2})$ as illustrated by the blue points in figure 2.3 and can be written as

$$\begin{aligned} f^*_{v_{i+\frac{1}{2},j}} &= \frac{1}{2}\left(f_{v_{i+\frac{1}{2},j+\frac{1}{2}}} + f_{v_{i+\frac{1}{2},j-\frac{1}{2}}}\right) \\ g^*_{v_{i+\frac{1}{2},j}} &= \frac{1}{2}\left(g_{v_{i+\frac{1}{2},j+\frac{1}{2}}} + g_{v_{i+\frac{1}{2},j-\frac{1}{2}}}\right). \end{aligned}$$

Next we define the auxiliary control volume for the discretization of the viscous terms. The blue rectangle in figure 2.3 is formed by joining the cell centers of the four cells

that share the cell vertex at $(i+\frac{1}{2}, j+\frac{1}{2})$. From equation (2.8) the viscous fluxes at the cell vertex can then be expanded for two-dimensional flow as follows

$$f_{v_{i+\frac{1}{2},j+\frac{1}{2}}} = \left\{ \begin{array}{c} 0 \\ \tau_{xx} \\ \tau_{xy} \\ u\tau_{xx} + v\tau_{xy} + k\frac{\partial T}{\partial x} \end{array} \right\}_{i+\frac{1}{2},j+\frac{1}{2}}$$

$$g_{v_{i+\frac{1}{2},j+\frac{1}{2}}} = \left\{ \begin{array}{c} 0 \\ \tau_{yx} \\ \tau_{yy} \\ u\tau_{yx} + v\tau_{yy} + k\frac{\partial T}{\partial y} \end{array} \right\}_{i+\frac{1}{2},j+\frac{1}{2}},$$

where the indices i and j in equation (2.8) have been replaced with x and y to avoid confusion with the i and j index used in this sub-section of the chapter. The $(i+\frac{1}{2}, j+\frac{1}{2})$ subscript is placed outside the braces to avoid repeating the subscript for every term within the braces. The velocity components, u and v, the coefficient of thermal conductivity, k, and temperature T are averaged between the values from the four cells that form the auxiliary control volume. The following is an example of how any one of these parameters are computed

$$u_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{1}{4}\left(u_{i,j} + u_{i+1,j} + u_{i,j+1} + u_{i+1,j+1}\right). \tag{2.25}$$

The viscous stress tensors, τ_{xx} and τ_{yy}, act in the horizontal and vertical directions, normal to the faces of the auxiliary control volume and are appropriately termed normal stresses. The shear stress, τ_{xy}, is exerted in the xy plane. The method by which the viscous stress tensors and the temperature gradients are calculated require the following procedure. The steps will only be shown for the normal stress tensor, τ_{xx}. The calculation for all other stress tensors will follow in a similar fashion. First, expand equation (2.10) with the appropriate, i and j subscripts

$$\tau_{xx_{i+\frac{1}{2},j+\frac{1}{2}}} = 2\mu_{i+\frac{1}{2},j+\frac{1}{2}}\left[\frac{\partial u}{\partial x}\right]_{i+\frac{1}{2},j+\frac{1}{2}} + \lambda_{i+\frac{1}{2},j+\frac{1}{2}}\left\{\left[\frac{\partial u}{\partial x}\right] + \left[\frac{\partial v}{\partial y}\right]\right\}_{i+\frac{1}{2},j+\frac{1}{2}}, \tag{2.26}$$

where the first and second coefficient of viscosities, μ and λ, are calculated by averaging the values between the four cells that form the auxiliary control volume as shown in equation (2.25). The first coefficient of viscosity, μ, is a combination of the laminar and turbulent viscosity coefficients defined as

$$\mu_{i+\frac{1}{2},j+\frac{1}{2}} = (\mu_{lam} + \mu_{turb})_{i+\frac{1}{2},j+\frac{1}{2}} .$$

The laminar coefficient of viscosity is calculated using the Sutherland equation defined in equation (2.11) and the eddy viscosity coefficient is calculated using the Baldwin-Lomax turbulence model [4] defined in equation (2.14). Then the velocity gradients are calculated by a transformation to the computational coordinates

$$\left[\frac{\partial u}{\partial x}\right]_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{1}{J_{i+\frac{1}{2},j+\frac{1}{2}}} \left\{ \left[\frac{\partial u}{\partial \xi}\right]_{i+\frac{1}{2},j+\frac{1}{2}} y_{\eta_{i+\frac{1}{2},j+\frac{1}{2}}} + \left[\frac{\partial u}{\partial \eta}\right]_{i+\frac{1}{2},j+\frac{1}{2}} y_{\xi_{i+\frac{1}{2},j+\frac{1}{2}}} \right\}. \quad (2.27)$$

Lastly, J is the volume of the auxiliary control volume and the velocity gradient in the computational domain, for an example, $\frac{\partial u}{\partial \xi}$, can be calculated by taking an average of the velocity differences in the x direction between the top and bottom faces of the auxiliary control volume as shown below

$$\left[\frac{\partial u}{\partial \xi}\right]_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{(u_{i+1,j+1} - u_{i,j+1}) + (u_{i+1,j} - u_{i,j})}{2}. \quad (2.28)$$

Note here that the metric terms in equation (2.27) differ from the metric terms used in the calculation of the flux velocity in equation (2.23) or the calculation of the viscous flux in equation (2.24), where in equation (2.27), the cell center coordinates are used instead of the cell vertex coordinates. Therefore, $y_{\eta_{i+\frac{1}{2},j+\frac{1}{2}}}$, is calculated by taking the average of the difference between the y coordinate in the η direction from the front and back faces of the auxiliary control volume. The formulation described in this sub-section to compute the viscous flux, guaranties conservation and produces a second order accurate algorithm.

2.2.2 Artificial Dissipation

The convective and viscous flux gradients are represented in discrete form using central second order spacial discretizations. All second order central schemes generate oscillations around discontinuities and allow odd-even point decoupling of the solution. To illustrate this point, let us consider the one-dimensional inviscid Navier-Stokes equation

$$\frac{\partial (Jw)_i}{\partial t} + \frac{\partial f}{\partial x} = 0.$$

Now apply the central second order spatial discretization to the flux gradient. Then the equation can be represented by a system of ordinary differential equations, such as

$$\frac{d(Jw)_i}{dt} = -\frac{f_{i+1} - f_{i-1}}{2\Delta x}.$$

Next solve the above equation on a uniform one-dimensional Cartesian mesh. Then a solution that takes on the value 1 on the odd cells, $(\ldots, i-3, i-1, i+1, i+3, \ldots)$, and the value -1 on the even cells, $(\ldots, i-2, i, i+2, \ldots)$ satisfies the above equation in the steady-state limit. This is referred to odd-even point decoupling. The error is of wavelength $2\Delta x$ and manifests itself as a high frequency error. In 1950 Von Neumann and Richtmyer [66] introduced the concept of artificial dissipation. The additional terms effectively coupled the terms in the equation and generate dissipation to eliminate oscillations around discontinuities and prevent expansion shocks at sonic transitions.

The artificial dissipation scheme used in this research is a blended first and third order flux, first introduced by Jameson et al. [43]. The artificial dissipation scheme can be defined as

$$d = \epsilon^{(2)} \Delta x^3 \frac{\lambda}{p} \left| \frac{\partial^2 p}{\partial x^2} \right| \frac{\partial w}{\partial x} - \epsilon^{(4)} \Delta x^3 \lambda \frac{\partial^3 w}{\partial x^3}, \tag{2.29}$$

where $\epsilon^{(2)}$ and $\epsilon^{(4)}$ are adjustable constants and λ is the spectral radius of the flux Jacobian matrices.

The first term in equation (2.29) is a first order flux term and is proportional to the normalized second difference of pressure. The pressure gradient term is negligible

in smooth regions of the flow but increases in magnitude around shocks, where the pressure field changes drastically. This has the effect of increasing the magnitude of dissipation around discontinuities. Unlike upwind schemes such as flux vector splitting and flux difference splitting, that consider the direction of the physical propagation of the waves, central schemes do not distinguish between the direction of upstream and downstream flow. But any upwind scheme can be represented as a combination of a central scheme with dissipation terms. Correspondingly, the introduction of the first order dissipative flux term introduces an upwind biasing to the second order central scheme and allows clean capture of shock waves.

In discrete form, the first order dissipation term can be written for the $(i+\frac{1}{2},j)$ face as

$$d^{(2)}_{i+\frac{1}{2},j} = \nu^{(2)}\Lambda_{i+\frac{1}{2},j}(w_{i+1,j} - w_{i,j}),$$

where

$$\Lambda_{i+\frac{1}{2},j} = \frac{1}{2}\left(\widetilde{\lambda}_{\xi_{i+1,j}} + \widetilde{\lambda}_{\xi_{i,j}}\right)$$

and $\widetilde{\lambda}_\xi$ and $\widetilde{\lambda}_\eta$ are the scaled spectral radii of the flux Jacobian matrices in the computational domain. The spectral radii are scaled to improve the convergence properties of the algorithm on stretched meshes. The scaling can be accomplished with the following formula

$$\widetilde{\lambda}_\xi = \left[1 + \left(\frac{\bar{\lambda}_\xi}{\bar{\lambda}_\eta}\right)^\sigma\right]\bar{\lambda}_\xi, \qquad \widetilde{\lambda}_\eta = \left[1 + \left(\frac{\bar{\lambda}_\eta}{\bar{\lambda}_\xi}\right)^\sigma\right]\bar{\lambda}_\eta,$$

where $\bar{\lambda}_\xi$ and $\bar{\lambda}_\eta$ are the spectral radii of the flux Jacobian matrices as defined in section 2.2.5. In practical computations $\sigma = \frac{2}{3}$ provides good results. The purpose of multiplying the first order dissipative flux with the spectral radius is to control the magnitude of the dissipative term. In viscous flow, the flow field velocity in the ξ direction is approximately equal to the freestream velocity in the far-field region but approaches zero within the boundary layer. The magnitude of the dissipative term should be much smaller than the magnitude of the viscous flux gradient. Otherwise it would introduce an excessive amount of dissipation into the flow field and corrupt the quality of the solution.

The $\nu^{(2)}$ term also controls the amount of dissipation throughout the flow field and is made proportional to the normalized second difference of the pressure field. The term acts as a pressure gradient sensor and turns on at regions with large pressure changes such as shock waves. The sensor can be defined as

$$\nu_{i,j}^{(2)} = \epsilon^{(2)} \max(\sigma_{i,j}, \sigma_{i+1,j}),$$

where

$$\sigma_{i,j} = \frac{|p_{i+1,j} - 2p_{i,j} + p_{i-1,j}|}{p_{i+1,j} + 2p_{i,j} + p_{i-1,j}}.$$

Unity is a typical value for $\epsilon^{(2)}$.

The third derivative term can be expressed as

$$d_{i+\frac{1}{2},j}^{(4)} = \nu^{(4)} \Lambda_{i+\frac{1}{2},j} \left(w_{i+2,j} - 3w_{i+1,j} + 3w_{i,j} - w_{i-1,j} \right),$$

where

$$\nu^{(4)} = \max \left[0, (\epsilon^{(4)} - \nu^{(2)}) \right].$$

The $\nu^{(4)}$ sensor is defined to turn off in regions of high pressure gradient since it has the tendency to reintroduce oscillations around discontinuities. The $\epsilon^{(4)}$ constant is defined such that the magnitude of $\nu^{(2)}$ exceeds it around discontinuities. The sensor $\nu^{(4)}$ then returns a value of zero around these regions and effectively turns off the third derivative term. A typical value for $\epsilon^{(4)}$ is $\frac{1}{32}$.

The complete first and third order dissipative flux can be formulated as

$$\begin{aligned} d_{i+\frac{1}{2},j} &= d_{i+\frac{1}{2},j}^{(2)} - d_{i+\frac{1}{2}.j}^{(4)} \\ d_{i+\frac{1}{2},j} &= \nu^{(2)} \Lambda_{i+\frac{1}{2},j} (w_{i+1,j} - w_{i,j}) \\ &\quad -\nu^{(4)} \Lambda_{i+\frac{1}{2},j} (w_{i+2,j} - 3w_{i+1,j} + 3w_{i,j} - w_{i-1,j}). \end{aligned} \tag{2.30}$$

2.2.3 Discrete Boundary Condition

In this sub-section, the boundary conditions discussed earlier for the field equations need to be cast in a discrete form. Additional constraints to that already defined

are sometimes required to ensure proper convergence of the numerical scheme used in this work. The following two sub-sections discuss the wall and far-field discrete boundary conditions.

Wall Boundary Conditions

Before we impose the inviscid and viscous wall boundary conditions, we first define the concept of "ghost" cells. For a cell-centered scheme, the flow variables are stored at the cell centers, thus in order to impose the boundary condition at the wall surface, "ghost" cells are introduced below the surface. Since the fluxes through the boundary faces must equal zero, then all convective terms are dropped and the flux F_2 from equation (2.17) can be defined as

$$F_2 = x_\xi p - y_\xi p,$$

where p is the pressure at the wall. Thus in order to impose the Euler wall boundary condition, only the value of pressure at the wall is needed. A simple way of calculating the pressure at the wall is to assume zeroth order extrapolation through the wall and thus copy the value of pressure from the cell above the wall to the ghost cell. In this work, the effect of the curvature of the streamlines is introduced by a method proposed by Rizzi [74]. The pressure gradient normal to the wall may be determined from the following expression

$$\left(x_\xi^2 + y_\xi^2\right) p_\eta = \left(x_\xi x_\eta + y_\xi y_\eta\right) p_\xi + \rho \left(y_\eta u - x_\eta v\right) \left(v x_{\xi\xi} - u y_{\xi\xi}\right).$$

The value of pressure at the wall is then calculated from the pressure gradient.

The "ghost" values for the velocity components are not needed but are computed in the code for completeness. The values are calculated for the "ghost" cells such that the velocity at the wall satisfies the boundary condition. For the inviscid case, since the normal component of the velocity vector is required to be zero, then the horizontal velocity component for the "ghost" cell termed as $\vec{u}_{i,1}$ can be defined as

$$\vec{u}_{i,1} = \vec{u}_{i,2} - 2(\vec{u} \cdot \vec{n})\vec{n}.$$

In the viscous case, the wall velocity must be zero, so the values of the velocity components below the wall take on the same magnitude as the ones above the wall but with opposite signs such as

$$u_{i,1} = -u_{i,2} \text{ and } v_{i,1} = -v_{i,2}.$$

Note that the "ghost" cells do not require an actual physical cell since its existence is only needed to calculate and store the flow variables such that the values at the wall satisfy the boundary condition. Therefore, the physical geometry and the corresponding metrics and cell volume are not calculated.

Far-Field Boundary Conditions

The method outlined here is based upon the procedure described by Jameson et al. [40]. The reader is urged to refer to this paper for a more detailed review.

At the far-field boundary, a characteristic based boundary condition using Riemann invariants are imposed. The method extrapolates the outward waves based on interior information and calculates incoming waves using the freestream conditions. Based on the one-dimensional Riemann invariant method, the outward and inward characteristics are calculated assuming that the flow is normal to the far-field boundary. The Riemann invariants can be expressed as

$$\begin{aligned} R_\infty &= v_\infty \cdot \vec{n} - \frac{2c_\infty}{\gamma - 1} \\ R_e &= v_e \cdot \vec{n} + \frac{2c_e}{\gamma - 1}, \end{aligned}$$

where the subscript ∞ signifies freestream values and e refers to the interior extrapolated values. The equations for the Riemann invariants can then be used to form expressions for the velocity normal to the boundary and its corresponding speed of

sound as

$$\begin{aligned} v_a \cdot \vec{n} &= \frac{1}{2}\left(R_\infty + R_e\right) \\ c_a &= \frac{\gamma - 1}{4}\left(-R_\infty + R_e\right). \end{aligned}$$

The next procedure is to calculate the velocity at the outflow and inflow boundary for the two-dimensional computational domain. Using a velocity triangle, the boundary velocity v_a can be defined as

$$v_a = v_e \cdot \vec{n}_t + v_a \cdot \vec{n},$$

where v_e is the extrapolated value and used for the tangential component since the Riemann invariant method is one-dimensional and only returns the velocity normal to the boundary. A similar procedure is used for the inflow boundaries and the velocity v_a can be expressed as

$$v_a = v_\infty \cdot \vec{n}_t + v_a \cdot \vec{n},$$

where v_∞ is the undisturbed freestream velocity. The choice of using the extrapolated tangential velocity from the interior at the outflow boundary and the tangential freestream velocity at the inflow boundary are the only natural options since its consistent with the direction of flow at these boundaries.

Finally, the entropy is extrapolated at the outflow boundary and the freestream entropy is used at the inflow boundary to calculate the values for density, pressure, and energy.

2.2.4 Time Stepping Scheme

The discretization of the Navier-Stokes equations described in the last few subsections in this chapter results in a set of coupled ordinary differential equations. A time integration scheme must be introduced to march the solution to a steady-state solution. In this work, we use the modified Runge-Kutta approach introduced by Jameson et al. [43].

The Runge-Kutta family of time integration techniques can provide a high order of accuracy for nonlinear equations, and have proved very effective in practice. A thorough examination of the classical Runge-Kutta method can be found in Hirsch [28], Gear [22], Lambert [48], and Van der Houwen [16]. Hirsch [28] described the Runge-Kutta method as a technique that evaluates the convective and dissipative flux gradients at several values of the state vector in the interval between $n\Delta t$ and $(n+1)\Delta t$ and combine them in order to obtain a high-order approximation of the state vector at the $(n+1)$ iteration. The Runge-Kutta method can be written as

$$\begin{aligned} W^{(0)} &= W^{(n)} \\ W^{(1)} &= W^{(0)} - \alpha_1 \Delta t R\left(W^{(0)}\right) \\ &\vdots \\ W^{(k)} &= W^{(0)} - \alpha_k \Delta t R\left(W^{(k-1)}\right) \quad \text{for } k = 1, 2, \cdots, M \\ W^{(n+1)} &= W^{(M)}, \end{aligned} \tag{2.31}$$

where M is the total number of stages.

The stability of the Runge-Kutta technique has been extensively studied. For hyperbolic problems, the maximum stability limit on the CFL number is $(M-1)$ along the imaginary axis for an M stage scheme. Solutions of the Navier-Stokes equations exhibit a combination of hyperbolic behavior in the main domain of the flow with parabolic behavior in regions close to the surface boundary where the viscous terms are dominant. Thus in addition to stability along the imaginary axis which is needed for an hyperbolic problem, stability along the real axis is also needed. The modified Runge-Kutta technique extends the stability region along the real axis by treating the convective and dissipative fluxes separately during each stage of the Runge-Kutta scheme. The residual at each stage of the scheme can be defined as

$$\begin{aligned} R^{(k)} &= C^{(k)} + D^{(k)} \\ C^{(k)} &= C\left(W^{(k)}\right) \\ D^{(k)} &= \beta_k D\left(W^{(k)}\right) + (1-\beta_k) D^{(k-1)}, \end{aligned}$$

where C and D are the convective and dissipative fluxes. The coefficients α_k are chosen to maximize the stability of the scheme along the imaginary axis and the coefficients β_k are chosen to maximize the stability of the scheme along the real axis. It turns out that the dissipative operators need not be evaluated at all stages of the scheme without any loss of stability. This ability of the modified Runge-Kutta, reduces the total cost of the scheme. In this work, a five stage Runge-Kutta scheme is used with three evaluations of the dissipative fluxes. The coefficients for the scheme are

$$\begin{array}{lllll} \alpha_1 = \frac{1}{4} & \alpha_2 = \frac{1}{6} & \alpha_3 = \frac{3}{8} & \alpha_4 = \frac{1}{2} & \alpha_5 = 1 \\ \beta_1 = 1. & \beta_2 = 0. & \beta_3 = 0.56 & \beta_4 = 0. & \beta_5 = 0.44. \end{array}$$

2.2.5 Convergence Acceleration

Over the years many new numerical tools have been developed to accelerate convergence and reduce the required time needed to produce a solution. The invention of faster computers has not eliminated the need for new approaches to produce faster results, since more complex problems are continually being attempted. In this subsection, we discuss three different convergence acceleration techniques used in this work.

Local Time Stepping

Increasing the time step would naturally accelerate the convergence of the time integration technique but is subject to a limit defined by the stability region. The limit can be calculated by taking the ratio of the cell volume to the sum of the spectral radii of the flux Jacobian matrices. For this purpose we need to determine the eigenvalues and spectral radii of the Jacobians appearing in the Euler and Navier-Stokes equations. Expanding the partial derivatives of the convective fluxes in equation (2.16) yields the following equation

$$\frac{\partial (Jw)_{ij}}{\partial t} + \widehat{A}(w)\frac{\partial w}{\partial \xi} + \widehat{B}(w)\frac{\partial w}{\partial \eta} = 0,$$

where $\xi = \xi_1$, $\eta = \xi_2$, and $\widehat{A}(w)$ and $\widehat{B}(w)$ are the flux Jacobians

$$\widehat{A}(w) = \frac{\partial F_1}{\partial w}, \quad \widehat{B}(w) = \frac{\partial F_2}{\partial w},$$

in the transformed coordinate system. Since $F_1 = y_\eta f - x_\eta g$, the flux Jacobian $\widehat{A}(w)$ can be written as

$$\widehat{A}(w) = y_\eta A(w) - x_\eta B(w).$$

The eigenvalues of matrix $\widehat{A}(w)$ are

$$\begin{aligned} \widehat{\lambda}_1 = \widehat{\lambda}_2 &= \vec{v} \cdot \vec{s}, \\ \widehat{\lambda}_3 &= \vec{v} \cdot \vec{s} + c||\vec{s}||, \\ \widehat{\lambda}_4 &= \vec{v} \cdot \vec{s} - c||\vec{s}||, \end{aligned}$$

where $\vec{v}$ is the velocity vector in each cell in the computational domain, c is the speed of sound, and $\vec{s}$ is the directional vector. The spectral radius is the largest eigenvalue of a given matrix and can be computed by the formula

$$\bar{\lambda} = \max\left(\widehat{\lambda}\right) = |\vec{v} \cdot \vec{s}| + c\|\vec{s}\|.$$

In the ξ direction, the directional vector $\vec{s}$ is defined as $\begin{bmatrix} y_\eta \\ -x_\eta \end{bmatrix}$. Then the spectral radius in the ξ direction for each cell can be expressed as

$$\bar{\lambda}_{\xi_{ij}} = |y_\eta u - x_\eta v|_{ij} + c_{ij}\left(y_{\eta_{ij}}^2 + x_{\eta_{ij}}^2\right).$$

And the spectral radius in the η direction for each cell is

$$\bar{\lambda}_{\eta_{ij}} = |-y_\xi u + x_\xi v|_{ij} + c_{ij}\left(y_{\xi_{ij}}^2 + x_{\xi_{ij}}^2\right).$$

The time step limit can now be estimated as

$$\Delta t^*_{ij} = \frac{\mathrm{CFL}}{\left(\bar{\lambda}_\xi + \bar{\lambda}_\eta\right)_{ij}}, \tag{2.32}$$

where $\bar{\lambda}_{\xi_{ij}}$ and $\bar{\lambda}_{\eta_{ij}}$ are the spectral radii of each cell, calculated using the local state vector and CFL depends on the stability region of the modified Runge-Kutta scheme. A conservative approach to calculating the time step limit would be to calculate the maximum spectral radius in the entire computational domain, and use a fixed time step throughout the domain which satisfies this limit. This would ensure that the scheme is stable and time accurate. The maximum stable time step is actually limited by the CFL condition, which states that the domain of dependence of the numerical scheme must at least contain the region of dependence of the differential equation and consequently it decreases with the local mesh interval. In a problem where the mesh size is almost equivalent throughout the computational domain, the use of a time step limit defined by the maximum spectral radius for the entire domain would be reasonable. However, for viscous problems, the ratio between cells close to the wall boundary and the far-field can amount to millions, and such an approach would require an excessive amount of iterations. One obvious approach to accelerate the convergence is to use a variable time step, where the state vector in each cell is advanced using a local time step satisfying the stability limit for that cell.

Nonlinear effects may predominate in regions of high pressure gradient. These include stagnations points, corners such as wing tips and trailing edges, and neighborhoods of shockwaves. Accordingly the scheme can be made more robust by reducing the time step in these regions. The time step used in this work can be expressed as

$$\Delta t_{ij} = \frac{\Delta t^*_{ij}}{\left(1 + \sigma_\xi + \sigma_\eta\right)_{ij}}, \tag{2.33}$$

where Δt^*_{ij} is from equation (2.32), and σ_ξ and σ_η are the pressure sensors or normalized second difference of the pressure field in each direction. In smooth regions of the flow, where no maximum or minimum in the pressure field exists, then the pressure sensors would return a value close to zero. Thus, the time step Δt is almost equal

to Δt^*. However, in regions of the flow especially around shock waves, the pressure sensors would return a positive value, and subsequently reduce the local time step.

For viscous problems, the time step derived in equation (2.32) needs to be augmented to include the maximum eigenvalue of the diffusion operator. Equation (2.32) can be expressed as

$$\Delta t^*_{ij} = \frac{CFL}{\left(\bar{\lambda}_\xi + \bar{\lambda}_\eta + \bar{\lambda}_{v_\xi} + \bar{\lambda}_{v_\eta}\right)_{ij}} , \tag{2.34}$$

where $\bar{\lambda}_{v_\xi}$ and $\bar{\lambda}_{v_\eta}$ are the estimates of the spectral radii of the viscous Jacobian fluxes and are defined as

$$\begin{aligned}
\bar{\lambda}_{v_\xi} &= \frac{\sqrt{\gamma} M_\infty}{\mathrm{Re} J}\left[\frac{\gamma\mu}{\mathrm{Pr}\rho}\left(x_\xi^2 + y_\xi^2\right) + \frac{\mu}{3\rho}\sqrt{\left(x_\xi^2 + y_\xi^2\right)\left(x_\eta^2 + y_\eta^2\right)}\right] \\
\bar{\lambda}_{v_\eta} &= \frac{\sqrt{\gamma} M_\infty}{\mathrm{Re} J}\left[\frac{\gamma\mu}{\mathrm{Pr}\rho}\left(x_\eta^2 + y_\eta^2\right) + \frac{\mu}{3\rho}\sqrt{\left(x_\xi^2 + y_\xi^2\right)\left(x_\eta^2 + y_\eta^2\right)}\right],
\end{aligned}$$

where Re and Pr are the Reynolds and Prandtl numbers of the flow and J is the area of the control volume. Please refer to Martinelli [53] for a derivation of the estimates of the eigenvalues of the viscous Jacobian matrices.

Residual Averaging

In the previous sub-section, we showed that by using the local spectral radius that appears in the denominator of equation (2.32), the local time step can be increased thus accelerating the convergence. In this sub-section, we explore the other alternative to increasing the time step and that is to increase the CFL number.

In the present case, the CFL value used in this work, is dictated by the stability analysis of the time integration scheme. One approach that provides a method to increase the CFL limit is residual averaging [40]. Here, the residual is replaced by a weighted average of the surrounding residuals. A simple cost effective approach is to explicitly obtain the new residual by averaging the three cells including the current cell in each direction with a smoothing parameter ϵ. However, this technique has one major flaw, where at values of $\epsilon \geq \frac{1}{4}$ the computation would return a zero value for the new residual for high frequency modes.

An implicit approach is not limited by this disadvantage, however, it is more costly. In a one-dimensional problem, the implicit residual averaging can be written as

$$-\epsilon\widehat{R}_{i-1} + (1-2\epsilon)\widehat{R}_i - \epsilon\widehat{R}_{i+1} = R_i.$$

The method produces an unconditionally stable scheme for any CFL number, if the smoothing parameter satisfies the following rule

$$\epsilon \geq \left[\left(\frac{\mathrm{CFL}}{\mathrm{CFL}^*}\right)^2 - 1\right],$$

where CFL^* is the maximum allowable Courant number for the scheme without implicit residual averaging and CFL is the new Courant number. In two-dimensional flows, the analysis to produce limits for the smoothing parameter is more tedious. To implicitly average the residual at (i, j), requires information from the surrounding cells in both directions. One approach of extending the above analysis to two-dimensions is to factorize the formula such as

$$\left(1 - \epsilon_\xi \partial^2_{\xi\xi}\right)\left(1 - \epsilon_\eta \partial^2_{\eta\eta}\right)\widehat{R} = R.$$

The solution of the above expression for the new residual only requires the inversion of tridiagonal matrices which can be performed cheaply when compared to the unfactored form. A study of the limits of the smoothing parameter can be found in Martinelli [53]. The study concludes that the smoothing parameters in each direction should satisfy the following inequalities.

$$\begin{aligned}
\epsilon_\xi &= \max\left\{\frac{1}{4}\left[\left(\frac{\mathrm{CFL}}{\mathrm{CFL}^*}\widetilde{\lambda}_\xi\right)^2 - 1\right], 0\right\} \\
\epsilon_\eta &= \max\left\{\frac{1}{4}\left[\left(\frac{\mathrm{CFL}}{\mathrm{CFL}^*}\widetilde{\lambda}_\eta\right)^2 - 1\right], 0\right\},
\end{aligned}$$

where $\widetilde{\lambda}_\xi$ and $\widetilde{\lambda}_\eta$ are the scaled spectral radii of the flux Jacobian matrices in each direction as defined in section 2.2.2, CFL is the new Courant number, and CFL^* is

the maximum allowable Courant number for the scheme without implicit residual averaging.

Multigrid Strategy

The multigrid method has proved to be the most effective convergence acceleration procedure of all. It was originally introduced by Fedorenko [18, 19] to accelerate the convergence of solutions governed by elliptic equations. Research by Brandt [8, 9] brought the multigrid concept to its full potential. Jameson [31, 32] perfected the technique for the solution of problems governed by hyperbolic equations.

In order to comprehend why the multigrid method is so successful, it is important to understand the properties of conventional iterative techniques. In order to reach equilibrium, information must be exchanged between every part of the domain. In an explicit time stepping scheme information is exchanged only between neighboring cells during each time step, or stage in the case of a Runge-Kutta scheme, with the consequence that high frequency errors are rapidly damped, but low frequency errors persist for many iterations. This leads to slow asymptotic convergence. A multigrid method achieves rapid exchange of information on the coarse grids. In the multigrid technique, the high frequency error modes are not visible in a coarser mesh but the low frequency error modes are well represented on the coarser meshes where they are viewed as high frequency modes. Therefore, at each subsequent coarse mesh, successively lower frequency modes from the finest mesh are reduced. The greater the number of coarser meshes, the larger the range of low frequency modes from the finest mesh that could be reduced and ultimately eliminated. In one multigrid cycle, almost the entire error frequency range is damped in comparison to conventional time integration techniques that damp the higher frequencies first and then progressively reduce the other frequencies. This has the additional benefit that it results in rapid convergence of global quantities of interest such as lift and drag.

When a structured mesh is used, successively coarser meshes can easily be generated simply by selecting every other point in each direction to obtain the next coarse mesh. This procedure can be continued until only two to three interior points exists in the coarsest mesh. With a cell centered scheme, this procedure results in

the agglomeration of 4 fine mesh cells in two-dimensions or 8 in three-dimensions to form each coarse mesh cell. For unstructured meshes, the procedure to obtain coarser meshes is not as obvious. This is still an open research topic.

The scheme used in this work is Jameson's multigrid time stepping scheme [31]. This was the first to use agglomerated cells for the coarse meshes. The general idea is to: first, perform one iteration of the time integration scheme on the Euler or Navier-Stokes equation on the finest mesh; second, transfer the solution and residual to the coarser mesh to obtain an approximation to the correction ΔW; third, transfer the corrections from the coarser mesh to the finer mesh to obtain a new approximation to the solution.

Alternative strategies for traversing the meshes can be derived. The simplest is a V-cycle, where the procedure traverses the multiple grids by descending through the levels to the coarsest mesh and then ascends the cycle until the finest mesh. The first step of the procedure is to apply the time integration method to obtain the first approximate solution for the finest mesh. Define the solution vector on the fine mesh as $w_h^{(0)}$.

Next, transfer the solution vector to the coarser mesh using a transfer operator. The solution vector on the coarser mesh can be defined as

$$w_H^{(0)} = T_h^H w_h^{(0)},$$

where T_h^H is the transfer operator. It is critical to the accuracy of the solution that the transfer operator preserves the conservative properties of the scheme. The transfer operator for a cell centered scheme used here can be defined as

$$w_H^{(0)} = \frac{\sum V_{ij} w_h^{(0)}}{\sum V_{ij}},$$

where V_{ij} is the cell volume of the fine grid cells. The next crucial step is to transfer a residual forcing function such that the solution vector on the coarse grid is primarily

driven by the transfered residuals from the fine grid during the first modified Runge-Kutta step. The forcing function can be defined as

$$P_H = Q_h^H R_h(w_h^{(0)}) - R_H(w_H^{(0)}),$$

where Q_h^H is the transfer operator for the fine grid residuals and can be obtained by simply adding the residuals from the fine grid cells that form the coarse grid cell as such

$$Q_h^H R_h(w_h^{(0)}) = \sum R_h(w_h^{(0)}).$$

The forcing function is then added to the coarse grid residual in the modified Runge-Kutta time stepping scheme as shown below.

$$\begin{aligned}
w_H^{(0)} &= w_H^{(n)} \\
w_H^{(1)} &= w_H^{(0)} - \alpha_1 \Delta t_H \left(R_H(w_H^{(0)}) + P_H \right) \\
&\vdots \\
w_H^{(k)} &= w_H^{(0)} - \alpha_k \Delta t_H \left(R_H(w_H^{(k-1)}) + P_H \right) \quad \text{for } k = 1, 2, \cdots, M \\
w_H^{(n+1)} &= w_H^{(M)}.
\end{aligned}$$

To show that the coarse grid solution vector is driven by the transfered fine grid residuals during the first Runge-Kutta time step, substitute the definition for the forcing function P_H into the second equation in the above algorithm.

$$w_H^{(1)} = w_H^{(0)} - \alpha_1 \Delta t_H \left(R_H(w_H^{(0)}) + Q_h^H R_h(w_h^{(0)}) - R_H(w_H^{(0)}) \right).$$

The coarse grid residuals cancel and only the transfered fine grid residuals remain. If further multigrid levels are desired, then the solution vector $w_H^{(M)}$ and its residual $R_H^{(M)}$ are transfered to the next coarse mesh. This completes the description of the descend process for a multigrid algorithm.

During the ascending process, the procedure is very simple. The basic idea is to update the solution vector at grid level h with the addition of the transfered difference

between the corrected and initial solution vector at grid level H. The process can be defined mathematically as

$$w_h^{new} = w_h + I_H^h \left(w_H^{new} - w_H^{(0)} \right).$$

This completes the detail description of the multigrid algorithm used in this work.

The V-cycle is the traditional procedure used to traverse the multigrid levels. The total computational work per multigrid cycle for the V-cycle satisfies the following bound

$$\text{Work}_{multigrid} < \frac{4}{3}\text{Work}_{finemesh}.$$

This estimate neglects the cost of computing the new grids at each level and the transfer operators, since these are negligible compared to the cost of computing the artificial dissipation and viscous fluxes. The W-cycle is an alternative and often the preferred route due to its ability to produce better convergence of the coarse grids before the solution vector is transfered to the fine grid. Since the low frequency errors generally require a longer amount of time to be eliminated, then the repeated transfer between the various coarse grids allows a greater reduction of the magnitude of the low frequency errors. The total work increases and is bounded by the following inequality

$$\text{Work}_{multigrid} < 2\text{Work}_{finemesh}.$$

The only disadvantage of the multigrid algorithm is the addition of high frequency errors into the solution vector in the fine mesh. However, if a good time integration routine is applied, then these high frequency errors would be eliminated. The α_k and β_k coefficients used in this work for the modified Runge-Kutta time stepping scheme have been optimized to provide good high frequency error damping.

Chapter 3

Numerical Optimization Algorithms

There are two main categories of optimization algorithms: non-gradient and gradient-based. In the first approach, only objective function evaluations are used to find the optimum point in the design space, and the gradient and the Hessian of the objective function are not needed. Hence the alternate name for this class of schemes "zeroth order methods". Some non-gradient methods have the advantage that they may be able to identify the global minimum. Unfortunately, however, they need large numbers of design cycles to obtain an optimum solution. Current algorithms of choice that belong to the non-gradient family of methods include: genetic algorithms, grid searches, stochastic, nonlinear simplex, etc. In genetic algorithms, evaluations of the figure of merit of an initial set of solutions starts the design process. This initial set is often very large: a typical characteristic of non-gradient-based methods. These algorithms handle both integer and continuous variables, in contrast to gradient-based methods that only work with the latter. During the conceptual design stage, the capability of optimization algorithms to handle integer variables allows the designer to choose, for example, the number of vertical tails, the number of engines, number of spars and other integer parameters. Another advantage of non-gradient-based methods is their ability to cope with objective functions that do not have smooth first or second derivatives.

The gradient-based approach has been widely used, and its application to aerodynamic design has a longer history than its counterpart. Gradient-based solvers such as SNOPT [24] and NPSOL [25] for non-linear programming have been extensively used. Gradient-based methods depend on the smoothness of the variation of the cost function with the design variables. This requires the existence of continuous first derivatives of the cost function; namely the gradient, and possibly higher derivatives. Simple gradient-based methods only require the gradient of the objective function with respect to the design variables. A prototype is the steepest descent method, which has a very low implementation cost, but usually requires a large number of iterations to converge to a local minimum, of the order of N^2 or more, where N is the number of design variables. More efficient methods such as Quasi-Newton methods, generate an approximation to the Hessian and generally only require N iterations before a rapid convergence to the local minimum is realized, since N iterations are needed to obtain a complete estimate of the Hessian. Line searches are usually introduced to find the minimum in the search direction at each step. Gradient-based methods generally require a much smaller number of iterations to converge to an optimum than non-gradient based methods. However, only convergence to a local minimum is guaranteed.

Both non-gradient and gradient-based methods have contrasting advantages and disadvantages. The key is to use a method that is appropriate for the mathematical model. In the conceptual design stage, where aircraft designers must decide on the number of engines, wing-tail instead of wing-canard combinations, various wing parameters, range, and payload, low-fidelity methods are often used to assess the designs. It is at this stage of an aircraft design process, where non-gradient-based methods can play a very important role. The use of low-fidelity methods, can drastically reduce the cost of each function evaluation and lower the turn-around time for new designs. These methods can be used to search for the global optimum, and they also allow integer parameters, features that are very beneficial to conceptual aircraft designers. Once the basic configuration is laid out, a gradient-based method can be used together with higher-fidelity flow solvers to refine the design. The appropriate optimization method also depends in the way in which the geometry is defined, as

will be discussed in section 4.7.

In this work, a smoothed steepest descent approach is applied, using gradient information provided by the adjoint method. In order to motivate this we begin with a brief discussion of the calculus of variations.

3.1 Calculus of Variations

In this section, the gradient and Hessian operators are discussed in relation to the calculus of variations. A detailed analysis is presented by Jameson and Vassberg [44]. First, we consider a class of optimization problems for which a curve $y(x)$ is to be chosen to minimize a cost function described by

$$I = \int_{x_0}^{x_1} F(x, y, y')\, dx,$$

where F is an arbitrary function that is continuous and twice-differentiable. The function F is dependent on x, y, and y', where $y(x)$ is the trajectory to be optimized and it is a continuous function and differentiable, and y' represents the derivative of y. Under a variation δy, the first variation of the cost function can be expressed as

$$\delta I = \int_{x_0}^{x_1} \left(\frac{\partial F}{\partial y} \delta y + \frac{\partial F}{\partial y'} \delta y' \right)\, dx,$$

Expand the equation by integrating the second term by parts

$$\delta I = \int_{x_0}^{x_1} \frac{\partial F}{\partial y} \delta y\, dx + \left[\frac{\partial F}{\partial y'} \delta y \right]_{x_0}^{x_1} - \int_{x_0}^{x_1} \frac{d}{dx} \frac{\partial F}{\partial y'} \delta y\, dx.$$

Assuming fixed end points, then the variations of y at x_0 and x_1 are zero, $\delta y(x_0) = \delta y(x_1) = 0$, so that

$$\delta I = \int_{x_0}^{x_1} \left(\frac{\partial F}{\partial y} - \frac{d}{dx} \frac{\partial F}{\partial y'} \right) \delta y\, dx = \int_{x_0}^{x_1} \mathcal{G} \delta y\, dx,$$

where $\mathcal{G}$ may be recognized as the gradient of the cost function and is expressed as

$$\mathcal{G} = \frac{\partial F}{\partial y} - \frac{d}{dx}\frac{\partial F}{\partial y'}.$$

A further variation of the gradient, then results to the following expression

$$\delta\mathcal{G} = \frac{\partial \mathcal{G}}{\partial y}\delta y + \frac{\partial \mathcal{G}}{\partial y'}\delta y' + \frac{\partial \mathcal{G}}{\partial y''}\delta y''$$

or

$$\delta\mathcal{G} = A\ \delta y,$$

where A is the Hessian operator. Thus the expression for the Hessian can be expressed as the differential operator

$$A = \frac{\partial \mathcal{G}}{\partial y} + \frac{\partial \mathcal{G}}{\partial y'}\frac{d}{dx} + \frac{\partial \mathcal{G}}{\partial y''}\frac{d^2}{dx^2}. \tag{3.1}$$

In the next sub-section, we explore this concept by deriving the gradient and Hessian operator for linearized supersonic flow [30].

3.1.1 Linearized Supersonic Flow

Consider a linearized supersonic flow over a profile with a height $y(x)$, where y is continuous and twice-differentiable. The surface pressure can be defined as

$$p - p_\infty = -\frac{\rho q^2}{\sqrt{M_\infty^2 - 1}}\frac{dy}{dx},$$

where $\frac{\rho q^2}{\sqrt{M_\infty^2-1}}$ is a constant and p_∞ is the freestream pressure [49]. Next consider an inverse problem with cost function

$$I = \frac{1}{2}\int_B (p - p_t)^2\ dx,$$

where p_t is the target surface pressure. The variation of the equation for the surface pressure and cost function under a profile variation δy is

$$\delta p = -\frac{\rho q^2}{\sqrt{M_\infty^2 - 1}} \frac{d}{dx} \delta y \quad \text{and} \quad \delta I = \int_B (p - p_t)\, \delta p \, dx.$$

Substitute the variation of the pressure into the equation for the variation of the cost function and integrate by parts to obtain

$$\begin{aligned} \delta I &= -\int_B (p - p_t) \frac{\rho q^2}{\sqrt{M_\infty^2 - 1}} \frac{d}{dx} \delta y \, dx \\ &= \int_B \left(\frac{\rho q^2}{\sqrt{M_\infty^2 - 1}} \right) \frac{d}{dx} (p - p_t) \delta y \, dx. \end{aligned}$$

The gradient can then be defined as

$$g = \frac{\rho q^2}{\sqrt{M_\infty^2 - 1}} \frac{d}{dx} (p - p_t).$$

To form the Hessian, take a variation of the gradient and substitute the expression for δp

$$\begin{aligned} \delta g &= \frac{\rho q^2}{\sqrt{M_\infty^2 - 1}} \frac{d}{dx} \delta p \\ &= -\frac{\rho^2 q^4}{M_\infty^2 - 1} \frac{d^2}{dx^2} \delta y. \end{aligned}$$

Thus the Hessian for the inverse design of the linearized supersonic flow problem can be expressed as the differential operator

$$A = -\frac{\rho^2 q^4}{M_\infty^2 - 1} \frac{d^2}{dx^2}. \tag{3.2}$$

3.2 Optimization Algorithms

This section addresses the steepest descent and the smoothed steepest descent optimization algorithms.

3.2.1 Steepest Descent

Line search methods require the algorithm to choose a direction p and search along this direction from the current iterate to obtain a new iterate for the function value. Once the direction is chosen, then a step length α is multiplied to the search direction to advance the optimization to the next iterate. In order to obtain the search direction, p, and the step length, α, we may employ Taylor's theorem. First, let us define the objective function as $f(x)$, then the optimization problem can be stated as

$$\min_x f(x),$$

where $x \in \mathbb{R}^n$ is a real vector with $n \geq 1$ components and $f\colon \mathbb{R}^n \to \mathbb{R}$ is a smooth function. Let, p be defined as the search direction. Then by Taylor's theorem

$$f(x + \alpha p) = f(x) + \alpha p^T \nabla f + \frac{1}{2}\alpha^2 p_T \nabla^2 f(x + tp)p + \dots \quad .$$

Here, the second term $p^T \nabla f$ is the rate of change of f along the search direction p. The last term contains the expression $\nabla^2 f(x + \alpha p)$ which corresponds to the Hessian matrix. The value p that would provide the most rapid decrease in the objective function $f(x)$, is the solution of the following optimization problem:

$$\min_p p^T \nabla f, \text{ subject to } \|p\| = 1.$$

With $\|p\| = 1$, the expression $p^T \nabla f$ can be simplified to

$$\begin{aligned} p^T \nabla f &= \|p\| \|\nabla f\| \cos\theta \\ &= \|\nabla f\| \cos\theta, \end{aligned}$$

where θ is the angle between the search direction p and the gradient ∇f. The above expression would attain its minimum value, when $\cos\theta$ takes on the value -1. Therefore, the equation can be further simplified to yield an expression for the search direction p of steepest descent

$$\begin{aligned} p^T \nabla f &= -\|\nabla f\| \\ p &= -\frac{\nabla f}{\|\nabla f\|}. \end{aligned}$$

Accordingly a simple optimization algorithm can then be defined by setting the search direction, p, to the negative of the gradient at every iteration. Therefore:

$$p = -\nabla f.$$

With a line search method the step size α is chosen such that the maximum reduction of the objective function $f(x)$ is attained. The vector x is then updated by the following expression:

$$x^{n+1} = x^n - \alpha \nabla f.$$

An alternative approach is to try to follow the continuous path of steepest descent in a sequence of many small steps. The equation above can be rearranged as such

$$\frac{x^{n+1} - x^n}{\alpha} = -\nabla f.$$

In the limit as $\alpha \to 0$, this reduces to

$$\frac{\partial x}{\partial t} = -\vec{\nabla} f, \tag{3.3}$$

where α is the time step in a forward Euler discretization. For the brachistochrone problem, Jameson and Vassberg [44] provide a stability limit, where the time step is dominated by the parabolic term in the continuous gradient formula. This leads to an estimate that the number of steps required to reach convergence is $\mathcal{O}\left(N^2\right)$.

3.2.2 Smoothed Steepest Descent

The search procedure used in this work is a descent method in which small steps are taken in a direction defined by the smoothed gradient. Let x represent the design variable, and ∇f the gradient. Instead of making the step

$$\delta x = \alpha p = -\alpha \nabla f,$$

we replace the gradient ∇f by a smoothed value $\overline{\nabla f}$. To apply smoothing in the x direction, the smoothed gradient $\overline{\nabla f}$ may be calculated from a discrete approximation to

$$\overline{\nabla f} - \frac{\partial}{\partial x}\epsilon\frac{\partial}{\partial x}\overline{\nabla f} = \nabla f, \tag{3.4}$$

where ϵ is the smoothing parameter. Then the first order change in the cost function is

$$\begin{aligned}
\delta f &= -\iint \nabla f \delta x dx \\
&= -\alpha \iint \left(\overline{\nabla f} - \frac{\partial}{\partial x}\epsilon\frac{\partial}{\partial x}\overline{\nabla f}\right)\overline{\nabla f} dx \\
&= -\alpha \iint \overline{\nabla f}^2 dx + \alpha \iint \left(\frac{\partial}{\partial x}\epsilon\frac{\partial}{\partial x}\overline{\nabla f}\right)\overline{\nabla f} dx.
\end{aligned}$$

Now, integrating the second integral by parts,

$$\begin{aligned}
\delta f &= -\alpha \iint \overline{\nabla f}^2 dx + \left[\alpha\overline{\nabla f}\epsilon\frac{\partial\overline{\nabla f}}{\partial x}\right] - \alpha \iint \epsilon\left(\frac{\partial\overline{\nabla f}}{\partial x}\right)^2 dx \\
&= -\alpha \iint \left(\overline{\nabla f}^2 + \epsilon\left(\frac{\partial\overline{\nabla f}}{\partial x}\right)^2\right) dx \\
&< 0,
\end{aligned}$$

where the second term in the first line of the equation is zero if the end points of the new gradient vector are assigned zero values. If the design variables are points on the surface of an airfoil and the objective function is some measure of the performance of the airfoil, then assigning zero values to the end points is analogous to fixing the

trailing edge points of the airfoil. If ϵ is positive, the variation of the objective function is less than zero and this assures an improvement if α is positive unless $\overline{\nabla f}$ and hence ∇f are zero. The smoothing ensures that each new shape in the optimization sequence remains smooth. It also acts as a preconditioner, which allows the use of much larger steps, and leads to a large reduction in the number of design iterations needed for convergence. A larger smoothing parameter allows a larger time step to be used and thus accelerates the convergence. Too large a smoothing parameter degrades the gradient information and can result in negligible modifications to the design variable and renders the implicit smoothing technique ineffective.

Jameson and Vassberg [44] show that the implicit smoothing technique corresponds to an implicit time stepping scheme for the descent equation (3.3) if the smoothing parameter ϵ is chosen appropriately. Consider a parabolic equation of the form

$$\frac{\partial x}{\partial t} = \pi \frac{\partial^2 x}{\partial y^2}.$$

A second order implicit discretization is

$$-\phi\delta x_{k-1} + (1+2\phi)\delta x_k - \phi\delta x_{k+1} = -\phi\left(x^n_{k-1} - 2x^n_k + x^n_{k+1}\right).$$

where $\phi = \frac{\pi\Delta t}{\Delta y^2}$. This corresponds exactly to smoothing the correction with the formula $\epsilon = \pi$. Their results show that the number of iterations required by the smoothing technique is similar to that of the implicit time stepping scheme, and both approaches perform better than the simple steepest descent and Quasi-Newton methods by a large amount.

For some problems, such as the calculus of variations, the implicit smoothing technique can be used to implement the Newton method. In a Newton method, the gradient is driven to zero based on the linearization

$$g(y + \delta y) = g(y) + A\delta y,$$

where A is the Hessian. In the case of the calculus of variations a Newton step can

Steepest Descent	N^2 steps
Quasi-Newton	N steps
Smoothed Gradient Steepest Descent	M steps (independent of N)

Table 3.1: Computational Cost of Gradient-Based Algorithms as a Function of the Number of Design Variables for the Brachistochrone Problem

be achieved by solving

$$A\delta y = \left(\frac{\partial \mathcal{G}}{\partial y} + \frac{\partial \mathcal{G}}{\partial y'} \frac{d}{dx} + \frac{\partial \mathcal{G}}{\partial y''} \frac{d^2}{dx^2} \right) \delta y = -g,$$

since the Hessian can be represented by the differential operator. Thus the correct choice of smoothing from equation (3.4) approximates the Newton step, resulting in quadratic convergence, independent of the number of mesh intervals.

In practice, this behavior has also been seen for aerodynamic shape optimization problems using the Euler and Navier-Stokes equations. Table 3.1 lists the computational cost of several gradient-based algorithms as a function of the number of design variables for the brachistochrone problem. Application of the Quasi-Newton method using softwares such as SNOPT [24] and NPSOL [25] for aerodynamic shape optimization has frequently produced converged solutions with far fewer iterations. Reuther et al. [70, 73] have shown drag minimization results which converge within 10-20 design cycles for three-dimensional applications.

Chapter 4

The Discrete and Continuous Adjoint Approaches

This chapter compares the continuous and discrete formulations of the adjoint method. The objective is to study the complexity of the discretization of the adjoint equation for both the continuous and discrete formulations, the accuracy of the resulting estimates of the gradient, and the impact on the computational cost to realize an optimum solution. First, the complete formulations and discretizations of the continuous and discrete viscous adjoint equations are presented. In particular, the differences between the continuous and discrete boundary conditions are explored. Second, the sensitivities obtained from continuous and discrete adjoint-based equations are compared to gradients derived by the finite-difference and complex-step methods. Finally, applications are presented for inverse, pressure and total drag minimization problems.

4.1 Formulation of the Optimal Design Problem

It is the intent of this work to investigate fully the derivation of both the continuous and discrete viscous adjoint methods. These results were first presented at the 29th AIAA Fluid Dynamics Conference, Albuquerque [39].

Aerodynamic optimization is based on the determination of the effect of shape

modifications on some performance measure that depends on the flow. These modifications alter the domain of the flow, which in this work is mapped to a fixed computational domain. For convenience, the coordinates ξ_i describing the fixed computational domain are chosen so that each boundary conforms to a constant value of one of these coordinates. Variations in the shape then result in corresponding variations in the mapping derivatives defined by K_{ij}.

Suppose that the performance is measured by a cost function

$$I = \int_{\mathcal{B}} \mathcal{M}\left(w, S\right) d\mathcal{B}_\xi + \int_{\mathcal{D}} \mathcal{P}\left(w, S\right) d\mathcal{D}_\xi \tag{4.1}$$

containing both boundary and field contributions, where $d\mathcal{B}_\xi$ and $d\mathcal{D}_\xi$ are the surface and volume elements in the computational domain. In general, $\mathcal{M}$ and $\mathcal{P}$ will depend on both the flow variables w and the metrics S defining the computational space. In the case of a multipoint design the flow variables may be separately calculated for several different conditions of interest.

The design problem is now treated as a control problem where the boundary shape represents the control function, which is chosen to minimize I subject to the constraints defined by the flow equations (2.15). A shape change produces a variation in the flow solution δw and the metrics δS, which in turn produce a variation in the cost function

$$\delta I = \int_{\mathcal{B}} \delta\mathcal{M}(w, S)\, d\mathcal{B}_\xi + \int_{\mathcal{D}} \delta\mathcal{P}(w, S)\, d\mathcal{D}_\xi, \tag{4.2}$$

with

$$\begin{aligned} \delta\mathcal{M} &= \left[\mathcal{M}_w\right]_I \delta w + \delta\mathcal{M}_{II}, \\ \delta\mathcal{P} &= \left[\mathcal{P}_w\right]_I \delta w + \delta\mathcal{P}_{II}, \end{aligned}$$

where subscripts I and II distinguish between the contributions associated with the variation of the flow solution δw and those associated with the metric variations δS. Thus $\left[\mathcal{M}_w\right]_I$ and $\left[\mathcal{P}_w\right]_I$ represent $\frac{\partial\mathcal{M}}{\partial w}$ and $\frac{\partial\mathcal{P}}{\partial w}$ with the metrics fixed, while $\delta\mathcal{M}_{II}$ and $\delta\mathcal{P}_{II}$ represent the contribution of the metric variations δS to $\delta\mathcal{M}$ and $\delta\mathcal{P}$. In the steady-state, the constraint equation (2.15) specifies the variation of the state vector

δw by

$$\frac{\partial}{\partial \xi_i}\delta\left(F_i - F_{v_i}\right) = 0. \tag{4.3}$$

Here δF_i and δF_{v_i} can also be split into contributions associated with δw and δS using the notation

$$\delta F_i = \left[F_{iw}\right]_I \delta w + \delta F_{iII}, \qquad \delta F_{v_i} = \left[F_{v_i w}\right]_I \delta w + \delta F_{v_i II}.$$

The inviscid contributions are easily evaluated as

$$\left[F_{iw}\right]_I = S_{ij}\frac{\partial f_j}{\partial w}, \quad \delta F_{iII} = \delta S_{ij} f_j.$$

The details of the viscous contributions are complicated by the additional level of derivatives in the stress and heat flux terms and are derived in Appendix A. Multiplying by a co-state vector ψ, also known as a Lagrange Multiplier, and integrating over the domain produces

$$\int_{\mathcal{D}} \psi^T \frac{\partial}{\partial \xi_i}\delta\left(F_i - F_{v_i}\right) = 0.$$

If ψ is differentiable this may be integrated by parts to give

$$\int_{\mathcal{B}} n_i \psi^T \delta\left(F_i - F_{v_i}\right) d\mathcal{B}_\xi - \int_{\mathcal{D}} \frac{\partial \psi^T}{\partial \xi_i}\delta\left(F_i - F_{v_i}\right) d\mathcal{D}_\xi = 0.$$

Since the left-hand expression equals zero, it may be subtracted from the variation of the cost function (4.2) to give

$$\delta I = \int_{\mathcal{B}} \left[\delta\mathcal{M} - n_i\psi^T\delta\left(F_i - F_{v_i}\right)\right] d\mathcal{B}_\xi + \int_{\mathcal{D}} \left[\delta\mathcal{P} + \frac{\partial\psi^T}{\partial\xi_i}\delta\left(F_i - F_{v_i}\right)\right] d\mathcal{D}_\xi. \tag{4.4}$$

Now, since ψ is an arbitrary differentiable function, it may be chosen in such a way that δI no longer depends explicitly on the variation of the state vector δw. The gradient of the cost function can then be evaluated directly from the metric variations without having to re-compute the variation δw resulting from the perturbation of each design variable.

The variation δw may be eliminated from (4.4) by equating all field terms with subscript "I" to produce a differential adjoint system governing ψ:

$$\frac{\partial \psi^T}{\partial \xi_i} \left[F_{iw} - F_{v_i w} \right]_I \; + \; \mathcal{P}_w = 0 \quad \text{in } \mathcal{D}. \tag{4.5}$$

The corresponding adjoint boundary condition is produced by equating the subscript "I" boundary terms in equation (4.4) to produce

$$n_i \psi^T \left[F_{iw} - F_{v_i w} \right]_I = \mathcal{M}_w \quad \text{on } \mathcal{B}. \tag{4.6}$$

The remaining terms from equation (4.4) then yield a simplified expression for the variation of the cost function that defines the gradient

$$\begin{aligned} \delta I \;\; = \;\; & \int_{\mathcal{B}} \left\{ \delta \mathcal{M}_{II} - n_i \psi^T \left[\delta F_i - \delta F_{v_i} \right]_{II} \right\} d\mathcal{B}_\xi \\ & + \int_{\mathcal{D}} \left\{ \delta \mathcal{P}_{II} + \frac{\partial \psi^T}{\partial \xi_i} \left[\delta F_i - \delta F_{v_i} \right]_{II} \right\} d\mathcal{D}_\xi. \end{aligned}$$

The details of the formula for the gradient depend on the way in which the boundary shape is parameterized as a function of the design variables, and the way in which the mesh is deformed as the boundary is modified. Using the relationship between the mesh deformation and the surface modification, the field integral is reduced to a surface integral by integrating along the coordinate lines emanating from the surface. Thus the expression for δI is finally reduced to

$$\delta I = \int_{\mathcal{B}} \mathcal{G} \delta \mathcal{F} \; d\mathcal{B}_\xi,$$

where $\mathcal{F}$ represents the design variables and $\mathcal{G}$ is the gradient, which is a function defined over the boundary surface.

The boundary conditions satisfied by the flow equations restrict the form of the left-hand side of the adjoint boundary condition (4.6). Consequently, the boundary contribution to the cost function $\mathcal{M}$ cannot be specified arbitrarily. Instead, it must be chosen from the class of functions that allow cancellation of all terms containing

δw in the boundary integral of equation (4.4). On the other hand, there is no such restriction on the specification of the field contribution to the cost function $\mathcal{P}$, since these terms can always be absorbed into the adjoint field equation (4.5) as source terms.

In the next two sections, a complete derivation of the continuous and discrete adjoint equations and boundary conditions are discussed for the case of two-dimensional flow. As in the case of the flow equations, the corresponding three-dimensional formulas can be derived in a similar manner. For simplicity, it is assumed that the portion of the boundary that undergoes shape modifications is restricted to the coordinate surface $\xi_2 = 0$. Then equations (4.4) and (4.6) may be simplified by incorporating the conditions

$$n_1 = 0, \quad n_2 = 1, \quad d\mathcal{B}_\xi = d\xi_1,$$

so that only the variations δF_2 and δF_{v_2} need be considered at the wall boundary.

4.2 Derivation of the Continuous Adjoint Terms

This section develops the continuous adjoint approach. Only the inviscid continuous adjoint terms are discussed. A detailed derivation of the viscous continuous adjoint equation and its corresponding boundary conditions for both the inverse design and drag minimization problems can be found in Appendix A.

The weak form of the Euler equations for steady flow is

$$\int_{\mathcal{D}} \frac{\partial \phi^T}{\partial \xi_k} F_k d\mathcal{D} = \int_{\mathcal{B}} n_k \phi^T F_k d\mathcal{B}, \tag{4.7}$$

where the test vector ϕ is an arbitrary differentiable function and n_k is the outward normal at the boundary. If a differentiable solution w is obtained to this equation, then it can be integrated by parts to give

$$\int_{\mathcal{D}} \phi^T \frac{\partial F_k}{\partial \xi_k} d\mathcal{D} = 0.$$

Since this is true for any ϕ, the differential form can be recovered. If the solution is

discontinuous, then equation (4.7) may be integrated by parts separately on either side of the discontinuity to recover the shock jump conditions.

Suppose now that we wish to control the surface pressure by varying the airfoil shape. For this purpose, it is convenient to retain a fixed computational domain. Variations in the shape then result in corresponding variations in the mapping derivatives defined by K. Introduce the cost function

$$I = \frac{1}{2}\int_{B_W} (p - p_d)^2 \, ds,$$

where p_d is the desired pressure. The design problem is now treated as a control problem where the control function is the airfoil shape, which is chosen to minimize I subject to the constrains defined by the flow equations (2.7). A variation in the shape causes a variation δp in the pressure and consequently a variation in the cost function

$$\delta I = \int_{B_W} (p - p_d)\, \delta p \;\; ds + \frac{1}{2}\int_{B_W} (p - p_d)^2 \;\; \delta ds.$$

Since p depends on w through the equation of state (2.9), the variation δp is determined from the variation δw. Define the Jacobian matrices as

$$A_k = \frac{\partial f_k}{\partial w}, \quad C_k = S_{kl} A_l. \tag{4.8}$$

The weak form of the equation for δw in the steady state becomes

$$\int_{\mathcal{D}} \frac{\partial \phi^T}{\partial \xi_k} \delta F_k d\mathcal{D} = \int_{\mathcal{B}} (n_k \phi^T \delta F_k) d\mathcal{B},$$

where

$$\delta F_k = C_k \delta w + \delta S_{kl} f_l,$$

which should hold for any differentiable test function ϕ.

This equation may be added to the variation in the cost function, which may now be written as

$$\begin{aligned}\delta I &= \int_{B_W} (p - p_d)\, \delta p \;\; ds + \frac{1}{2} \int_{B_W} (p - p_d)^2 \;\; \delta ds \\ &\quad - \int_{\mathcal{D}} \frac{\partial \psi^T}{\partial \xi_k} \delta F_k d\mathcal{D} + \int_{\mathcal{B}} (n_k \psi^T \delta F_k) d\mathcal{B}.\end{aligned}$$

On the airfoil surface B_W, $n_1 = 0$. Thus, it follows from equation (4.2) that

$$\delta F_2 = \begin{bmatrix} 0 \\ S_{21}\delta p \\ S_{22}\delta p \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \delta S_{21} p \\ \delta S_{22} p \\ 0 \end{bmatrix}.$$

Since the weak equation for δw should hold for an arbitrary choice of the test vector ϕ, we are free to choose ϕ to simplify the resulting expressions. Therefore we set $\phi = \psi$, where the costate vector ψ is the solution of the adjoint equation

$$\frac{\partial \psi}{\partial t} - C_k^T \frac{\partial \psi}{\partial \xi_k} = 0 \quad \text{in } D. \tag{4.9}$$

At the outer boundary, incoming characteristics for ψ correspond to outgoing characteristics for δw. Consequently we can choose boundary conditions for ψ such that

$$n_k \psi^T C_k \delta w = 0.$$

If the coordinate transformation is such that δS is negligible in the far-field, then the only remaining boundary term is

$$- \int_{B_W} \psi^T \delta F_2 \; d\xi_1.$$

Thus, by letting ψ satisfy the boundary condition,

$$\psi_j n_j = p - p_d \;\; \text{on } B_W, \tag{4.10}$$

where n_j are the components of the surface normal,

$$n_j = \frac{S_{2j}}{\sqrt{S_{2j}S_{2j}}},$$

we find finally that

$$\delta I = \frac{1}{2}\int_{B_W} (p - p_d)^2 \, \delta ds - \int_{\mathcal{D}} \frac{\partial \psi^T}{\partial \xi_k} \delta S_{kl} f_l d\mathcal{D} - \int_{B_W} (\delta S_{21}\psi_2 + \delta S_{22}\psi_3)\, p \, d\xi_1.$$

4.2.1 Numerical Discretization

The convective adjoint flux is discretized using a second order central spatial discretization. The first step is to expand equation (4.9) for a two-dimensional problem

$$\frac{\partial \psi}{\partial t} - C_1^T \frac{\partial \psi}{\partial \xi_1} - C_2^T \frac{\partial \psi}{\partial \xi_2} = 0.$$

Define $\xi = \xi_1$ and $\eta = \xi_2$. Then the continuous adjoint fluxes can be discretized as

$$\begin{aligned} V\frac{\partial \psi_{i,j}}{\partial t} &= \frac{1}{2}\left[C^T_{1_{i,j}}\left(\psi_{i+1,j} - \psi_{i-1,j}\right) + C^T_{2_{i,j}}\left(\psi_{i,j+1} - \psi_{i,j-1}\right)\right] \\ &\quad + d_{i+\frac{1}{2},j} - d_{i-\frac{1}{2},j} + d_{i,j+\frac{1}{2}} - d_{i,j-\frac{1}{2}}, \end{aligned}$$

where V is the cell area and $d_{i+\frac{1}{2},j}$ has the same form as equation (2.30). From equation (4.8), the Jacobian fluxes can be expanded as

$$\begin{aligned} C^T_{1_{i,j}} &= y_{\eta_{i,j}} A^T_{1_{i,j}} - x_{\eta_{i,j}} A^T_{2_{i,j}} \\ C^T_{2_{i,j}} &= -y_{\xi_{i,j}} A^T_{1_{i,j}} + x_{\xi_{i,j}} A^T_{2_{i,j}}, \end{aligned}$$

where

$$y_{\eta_{i,j}} = \frac{1}{2}\left(y_{\eta_{i+\frac{1}{2},j}} + y_{\eta_{i-\frac{1}{2},j}}\right), \quad x_{\eta_{i,j}} = \frac{1}{2}\left(x_{\eta_{i+\frac{1}{2},j}} + x_{\eta_{i-\frac{1}{2},j}}\right),$$

$$A^T_{1_{i,j}} = \left[\frac{\partial f}{\partial w}\right]^T, \quad A^T_{2_{i,j}} = \left[\frac{\partial g}{\partial w}\right]^T.$$

In order to reduce the number of subscripts and simplify the notation, the Euler Jacobian matrices are defined as follows

$$\widehat{A^T}_{i,j} = C^T_{1_{i,j}}, \quad \widehat{B^T}_{i,j} = C^T_{2_{i,j}}, \quad A^T_{i,j} = A^T_{1_{i,j}}, \quad B^T_{i,j} = A^T_{2_{i,j}}. \tag{4.11}$$

Finally, the convective continuous adjoint flux can be written as

$$\mathcal{R}(\psi) = \frac{1}{2}\left[\widehat{A^T}_{i,j}\left(\psi_{i+1,j} - \psi_{i-1,j}\right) + \widehat{B^T}_{i,j}\left(\psi_{i,j+1} - \psi_{i,j-1}\right)\right]. \tag{4.12}$$

4.2.2 Continuous Adjoint Boundary Conditions

In this sub-section the inviscid continuous adjoint boundary conditions are presented for the inverse and drag minimization problems.

Inverse Design

In the case of the continuous adjoint boundary condition, equation (4.10) constrains the values of the normal adjoint velocities. The tangential adjoint velocity, ψ_1, and ψ_4 do not appear; therefore, assigning a zero value for these variables does not violate equation (4.10). This results, however, in poor convergence for the adjoint equation because it is an over-specification of the adjoint boundary condition. A satisfactory boundary condition may be formulated as follows:

$$\begin{aligned}
\psi_{1_{i,1}} &= \psi_{1_{i,2}} \\
\psi_{2_{i,1}} &= \psi_{2_{i,2}} + 2n_2\left((p - p_d) - n_2\psi_{2_{i,2}} + n_1\psi_{3_{i,2}}\right) \\
\psi_{3_{i,1}} &= \psi_{3_{i,2}} - 2n_1\left((p - p_d) - n_2\psi_{2_{i,2}} + n_1\psi_{3_{i,2}}\right) \\
\psi_{4_{i,1}} &= \psi_{4_{i,2}},
\end{aligned} \tag{4.13}$$

where $n_i = \frac{S_{2i}}{\sqrt{S_{2j}S_{2j}}}$. The subscripts $(i,1)$ and $(i,2)$ in the above equations denote cells below and above the wall. Here, the first and fourth costate variables below the wall are set equal to the corresponding values above the wall and the tangential adjoint velocities above and below the wall are equated.

Drag Minimization

If the drag is to be minimized, then the cost function is the drag coefficient,

$$I = C_d = \left(\frac{1}{c}\int_{B_W} C_p \frac{\partial y}{\partial \xi} d\xi\right)\cos\alpha + \left(\frac{1}{c}\int_{B_W} -C_p \frac{\partial x}{\partial \xi} d\xi\right)\sin\alpha.$$

A variation in the shape causes a variation δp in the pressure and consequently a variation in the cost function,

$$\begin{aligned}\delta I &= \frac{1}{c}\int_{B_W} C_p \left(\frac{\partial y}{\partial \xi}\cos\alpha - \frac{\partial x}{\partial \xi}\sin\alpha\right)\delta p d\xi \\ &\quad + \frac{1}{c}\int_{B_W} C_p \left(\delta\left(\frac{\partial y}{\partial \xi}\right)\cos\alpha - \delta\left(\frac{\partial x}{\partial \xi}\right)\sin\alpha\right) d\xi.\end{aligned}$$

As in the inverse design case, the first term is a function of the state vector, and therefore is incorporated into the boundary condition, where the integrand replaces the pressure difference term in equation (4.13). The second term is added to the gradient term.

4.3 Derivation of the Discrete Adjoint Terms

In the continuous adjoint approach, the primary steps that are required to formulate the continuous adjoint equation are: first, derive the first variation of the flux gradient as shown in equation (4.3); second, multiply it by the Lagrange Multiplier and integrate over the domain; third, subtract the integral obtained from the second step from the variation of the cost function; fourth, perform integration by parts to isolate the variation of the state vector terms, δw, from the variation of the shape function, δf, to produce the continuous adjoint equations. The discrete adjoint equation is obtained by applying control theory directly to the set of discrete field equations, following the same sequence of steps. The resulting equations depend on the details of the scheme used to solve the flow equations.

To formulate the discrete adjoint equation, we first take a variation of the residual

term. From equation (2.20), the first variation can be written as

$$\delta R(w)_{ij} = \delta h_{i+\frac{1}{2},j} - \delta h_{i-\frac{1}{2},j} + \delta h_{i,j+\frac{1}{2}} - \delta h_{i,j-\frac{1}{2}}, \tag{4.14}$$

with

$$\begin{aligned} \delta h_{i+\frac{1}{2},j} &= \delta f_{i+\frac{1}{2},j} - \delta f_{v_{i+\frac{1}{2},j}} - \delta d_{i+\frac{1}{2},j} \\ \delta h_{i,j+\frac{1}{2}} &= \delta g_{i,j+\frac{1}{2}} - \delta g_{v_{i,j+\frac{1}{2}}} - \delta d_{i,j+\frac{1}{2}}, \end{aligned} \tag{4.15}$$

where f and g are the convective flux gradients, f_v and g_v are the viscous flux gradients, and d is the artificial dissipation term. Next, we pre-multiply the variation of the discrete residual by the Lagrange Multiplier and sum the product over the computational domain to produce the following

$$\sum_{i=2}^{nx}\sum_{j=2}^{ny} \psi_{i,j}^T \delta R(w)_{i,j}. \tag{4.16}$$

Thirdly, similarly to the the primary steps taken to produce the continuous adjoint equation, equation (4.16) is added to the variation of the discrete cost function,

$$\delta I = \delta I_c + \sum_{i=2}^{nx}\sum_{j=2}^{ny} \psi_{i,j}^T \delta R(w)_{i,j},$$

where δI_c is the discrete cost function, and $R(w)_{i,j}$ is the residual term.

In the continuous adjoint approach, integration by parts is employed to formulate the continuous adjoint partial differential system of equations. However, to develop the discrete adjoint equations, the discrete counterpart to the integration by parts, summation by parts is required. To produce the final set of discrete adjoint equations, expand the $\delta R(w)$ term for cell (i, j) and the adjacent four cells. Then multiply the variation of the residual by the Lagrange multiplier, $\psi_{i,j}$. Lastly, collect any term that is multiplied by $\delta w_{i,j}$. A full discretization of the equation would involve discretizing every term that is a function of the state vector. The above steps prove to be a daunting task, due to the need to keep track of the contribution from various terms

and cells. The development cost of the method grows rapidly with the order and size of the stencil of the discretization scheme.

In this work, the values of the flow variables are stored at the cell centers. The fluxes vary linearly between adjacent cells. The convective flux $f_{i+\frac{1}{2},j}$ at the cell face can be computed by taking the average of the flux contributions from each cell across the cell face. The total flux balance for each cell requires information from the four adjacent cells. In the case of the artificial dissipative flux, a blended first and third order flux term is used. The third order term requires information from two cells in each direction, therefore requiring information from a total of nine cells. A numerical evaluation of the viscous flux requires an estimate of the partial derivative of velocity in the viscous stress tensor and the partial derivative of temperature from the Fourier Law of heat conduction. To evaluate the viscous flux at the cell face, we first compute the stress tensor and the heat flux components of the viscous flux at the end points (vertex) of the edge by employing a discrete Gauss theorem to the auxiliary control volume formed by the centers of the four cells enclosing the vertex. Second, the viscous flux across the cell face is computed by averaging the viscous fluxes at both ends of the edge. Thus to develop the discrete viscous adjoint flux requires information from the eight surrounding cells.

As can be seen from the above example, the development cost of the discrete adjoint equation increases rapidly with the order of the numerical scheme. The development of the discrete adjoint for higher order schemes with large stencils, typically used for DNS and LES simulations or aeroacoustic calculations, would require a large amount of time and be more susceptible to errors during the formulation phase. The continuous adjoint approach might then be preferred, since its formulation is completely independent of the numerical scheme used to solve the flow field equations. If desired, a simpler numerical scheme that requires a smaller stencil could be used to compute the continuous adjoint equation, while a higher order scheme is used for the flow field equations. The next three sub-sections illustrate the development of the contributions from the convective, dissipative, and viscous flux terms to the discrete adjoint equation.

4.3.1 Contributions from the Convective Flux

In this sub-section we will concentrate our efforts on the contribution from the convective flux of the field equations. The numerical scheme we employ to solve the flow field equations utilize a central second order spatial discretization to evaluate the flux gradients in each direction. The fluxes are averaged at the flux faces before the flux gradients are computed. This is equivalent to a three-point stencil to evaluate the flux gradient in each direction.

First, we will only consider the contribution from $\delta f_{i+\frac{1}{2},j}$ to equation (4.15) and ultimately to equation (4.14). From equation (2.22), the first variation of the convective flux computed at the flux face can be written as

$$\delta f_{i+\frac{1}{2},j} = \frac{1}{2}\left(\delta f^{+}_{i+1,j} + \delta f^{-}_{i,j}\right). \tag{4.17}$$

Now expand $\delta f^{-}_{i,j}$,

$$\begin{aligned}
\delta f^{-}_{i,j} &= \delta\left[y_{\eta_{i+\frac{1}{2},j}} f_{i,j} - x_{\eta_{i+\frac{1}{2},j}} g_{i,j}\right] \\
&= y_{\eta_{i+\frac{1}{2},j}} \delta f_{i,j} + \delta y_{\eta_{i+\frac{1}{2},j}} f_{i,j} - x_{\eta_{i+\frac{1}{2},j}} \delta g_{i,j} - \delta x_{\eta_{i+\frac{1}{2},j}} g_{i,j} \\
&= y_{\eta_{i+\frac{1}{2},j}} \left(\frac{\partial f}{\partial w}\delta w\right)_{i,j} + \delta y_{\eta_{i+\frac{1}{2},j}} f_{i,j} - x_{\eta_{i+\frac{1}{2},j}} \left(\frac{\partial g}{\partial w}\delta w\right)_{i,j} - \delta x_{\eta_{i+\frac{1}{2},j}} g_{i,j} \\
&= \left(y_{\eta_{i+\frac{1}{2},j}} A_{i,j} - x_{\eta_{i+\frac{1}{2},j}} B_{i,j}\right)\delta w_{i,j} + \delta y_{\eta_{i+\frac{1}{2},j}} f_{i,j} - \delta x_{\eta_{i+\frac{1}{2},j}} g_{i,j},
\end{aligned} \tag{4.18}$$

where A and B are the convective flux Jacobians $\frac{\partial f}{\partial w}$ and $\frac{\partial g}{\partial w}$. Similarly, the $\delta f^{+}_{i+1,j}$ term can be expanded to produce the following

$$\delta f^{+}_{i+1,j} = \left(y_{\eta_{i+\frac{1}{2},j}} A_{i+1,j} - x_{\eta_{i+\frac{1}{2},j}} B_{i+1,j}\right)\delta w_{i+1,j} + \delta y_{\eta_{i+\frac{1}{2},j}} f_{i+1,j} - \delta x_{\eta_{i+\frac{1}{2},j}} g_{i+1,j}. \tag{4.19}$$

Note here that the metric terms in equation (4.18) and (4.19) are identical since the plus and minus fluxes are evaluated along the $(i+\frac{1}{2}, j)$ edge. The only difference between the two equations are the state vector terms. Substituting of equations (4.18) and (4.19) into equation (4.17), keeping only terms that are multiplied to the variation of the state vector and neglecting the variation of the dissipative and viscous

fluxes for now, equation (4.15) can be expanded to

$$\begin{aligned}\delta h_{i+\frac{1}{2},j} &= \frac{1}{2}\left[\left(y_{\eta_{i+\frac{1}{2},j}}A_{i+1,j} - x_{\eta_{i+\frac{1}{2},j}}B_{i+1,j}\right)\delta w_{i+1,j}\right. \\ &\quad \left.+\left(y_{\eta_{i+\frac{1}{2},j}}A_{i,j} - x_{\eta_{i+\frac{1}{2},j}}B_{i,j}\right)\delta w_{i,j}\right]. \end{aligned} \tag{4.20}$$

In the η direction, $\delta g_{i,j+\frac{1}{2}}$ can be written as

$$\delta g_{i,j+\frac{1}{2}} = \frac{1}{2}\left(\delta g^{+}_{i,j+1} + \delta g^{-}_{i,j}\right),$$

where

$$\begin{aligned}\delta g^{-}_{i,j} &= \delta\left[-y_{\xi_{i,j+\frac{1}{2}}}f_{i,j} + x_{\xi_{i,j+\frac{1}{2}}}g_{i,j}\right] \\ &= \left(-y_{\xi_{i,j+\frac{1}{2}}}A_{i,j} + x_{\xi_{i,j+\frac{1}{2}}}B_{i,j}\right)\delta w_{i,j} - \delta y_{\xi_{i,j+\frac{1}{2}}}f_{i,j} + \delta x_{\xi_{i,j+\frac{1}{2}}}g_{i,j}, \\ \delta g^{+}_{i,j+\frac{1}{2}} &= \left(-y_{\xi_{i,j+\frac{1}{2}}}A_{i,j+1} + x_{\xi_{i,j+\frac{1}{2}}}B_{i,j+1}\right)\delta w_{i,j+1} - \delta y_{\xi_{i,j+\frac{1}{2}}}f_{i,j+1} + \delta x_{\xi_{i,j+\frac{1}{2}}}g_{i,j+1}.\end{aligned}$$

Then $\delta h_{i,j+\frac{1}{2}}$ can be expressed as

$$\begin{aligned}\delta h_{i,j+\frac{1}{2}} &= \frac{1}{2}\left[\left(-y_{\xi_{i,j+\frac{1}{2}}}A_{i,j+1} + x_{\xi_{i,j+\frac{1}{2}}}B_{i,j+1}\right)\delta w_{i,j+1}\right. \\ &\quad \left.+\left(-y_{\xi_{i,j+\frac{1}{2}}}A_{i,j} + x_{\xi_{i,j+\frac{1}{2}}}B_{i,j}\right)\delta w_{i,j}\right]. \end{aligned} \tag{4.21}$$

We now have all the necessary terms to formulate the variation of the convective flux $\delta R(w)_{i,j}$. Substitution of equation (4.20) and (4.21) into equation (4.14) will produce

$$\delta R(w)_{i,j} = \delta h_{i+\frac{1}{2},j} - \delta h_{i-\frac{1}{2},j} + \delta h_{i,j+\frac{1}{2}} - \delta h_{i,j-\frac{1}{2}} \tag{4.22}$$

$$
\begin{aligned}
&\delta R(w)_{i,j} = \\
&\frac{1}{2}\left[\left(y_{\eta_{i+\frac{1}{2},j}}A_{i+1,j} - x_{\eta_{i+\frac{1}{2},j}}B_{i+1,j}\right)\delta w_{i+1,j} + \left(y_{\eta_{i+\frac{1}{2},j}}A_{i,j} - x_{\eta_{i+\frac{1}{2},j}}B_{i,j}\right)\delta w_{i,j}\right] \\
-\ &\frac{1}{2}\left[\left(y_{\eta_{i-\frac{1}{2},j}}A_{i-1,j} - x_{\eta_{i-\frac{1}{2},j}}B_{i-1,j}\right)\delta w_{i-1,j} + \left(y_{\eta_{i-\frac{1}{2},j}}A_{i,j} - x_{\eta_{i-\frac{1}{2},j}}B_{i,j}\right)\delta w_{i,j}\right] \\
+\ &\frac{1}{2}\left[\left(-y_{\xi_{i,j+\frac{1}{2}}}A_{i,j+1} + x_{\xi_{i,j+\frac{1}{2}}}B_{i,j+1}\right)\delta w_{i,j+1} + \left(-y_{\xi_{i,j+\frac{1}{2}}}A_{i,j} + x_{\xi_{i,j+\frac{1}{2}}}B_{i,j}\right)\delta w_{i,j}\right] \\
-\ &\frac{1}{2}\left[\left(-y_{\xi_{i,j-\frac{1}{2}}}A_{i,j} + x_{\xi_{i,j-\frac{1}{2}}}B_{i,j}\right)\delta w_{i,j} + \left(-y_{\xi_{i,j-\frac{1}{2}}}A_{i,j-1} + x_{\xi_{i,j-\frac{1}{2}}}B_{i,j-1}\right)\delta w_{i,j-1}\right].
\end{aligned}
$$

Simplify equation (4.22) and reorder the terms to produce an equation for the contribution of the convective flux from the field equations to the variation of the residual in cell (i, j),

$$
\begin{aligned}
\delta R(w)_{i,j} \ = \ &\frac{1}{2}\left[\left(y_{\eta_{i+\frac{1}{2},j}}A_{i+1,j} - x_{\eta_{i+\frac{1}{2},j}}B_{i+1,j}\right)\delta w_{i+1,j}\right. \\
&- \left(y_{\eta_{i-\frac{1}{2},j}}A_{i-1,j} - x_{\eta_{i-\frac{1}{2},j}}B_{i-1,j}\right)\delta w_{i-1,j} \\
&+ \left(\left(y_{\eta_{i+\frac{1}{2},j}} - y_{\eta_{i-\frac{1}{2},j}} - y_{\xi_{i,j+\frac{1}{2}}} + y_{\xi_{i,j-\frac{1}{2}}}\right)A_{i,j}\right. \\
&\qquad \left.- \left(x_{\eta_{i+\frac{1}{2},j}} - x_{\eta_{i-\frac{1}{2},j}} - x_{\xi_{i,j+\frac{1}{2}}} + x_{\xi_{i,j-\frac{1}{2}}}\right)B_{i,j}\right)\delta w_{i,j} \\
&+ \left(-y_{\xi_{i,j+\frac{1}{2}}}A_{i,j+1} + x_{\xi_{i,j+\frac{1}{2}}}B_{i,j+1}\right)\delta w_{i,j+1} \\
&\left.- \left(-y_{\xi_{i,j-\frac{1}{2}}}A_{i,j-1} + x_{\xi_{i,j-\frac{1}{2}}}B_{i,j-1}\right)\delta w_{i,j-1}\right]. \qquad (4.23)
\end{aligned}
$$

Note that the equation above has contributions from all four adjacent cells. In order to simplify the notation, $\delta R(w)_{i,j}$ will be represented by $\delta R_{i,j}$.

The variation of the residual vector from the adjacent cells such as $\delta R_{i+1,j}$, $\delta R_{i-1,j}$, etc., have contributions from the (i, j) cell. For example, if equation (4.23) is written for cell $(i+1, j)$ and only the $\delta w_{i,j}$ terms are shown, then the variation of the residual vector for cell $(i + 1, j)$ can be written as

$$
\delta R(w)_{i+1,j} = \cdots + \frac{1}{2}\left[-\left(y_{\eta_{i+\frac{1}{2},j}}A_{i,j} - x_{\eta_{i+\frac{1}{2},j}}B_{i,j}\right)\delta w_{i,j}\right] + \cdots
$$

The next step is to pre-multiply the variation of the residual vector by the transpose of the Lagrange multiplier vector and sum the product over the entire domain. This

step leads to the following equation

$$\begin{aligned}\sum_{i=2}^{nx}\sum_{j=2}^{ny}\psi_{i,j}^{T}\delta R_{i,j} &= \cdots+\psi_{i-1,j}^{T}\delta R_{i-1,j}+\psi_{i+1,j}^{T}\delta R_{i+1,j}+\psi_{i,j}^{T}\delta R_{i,j}\\ &\quad+\psi_{i,j-1}^{T}\delta R_{i,j-1}+\psi_{i,j+1}^{T}\delta R_{i,j+1}+\cdots . \end{aligned}\tag{4.24}$$

The discrete domain spans from $i=2\rightarrow nx$ and $j=2\rightarrow ny$, where nx and ny are the maximum cell points. Next we substitute the expansions for the variation of the residual terms from equation (4.23) for each term in equation (4.24) and collect the $\delta w_{i,j}$ terms to produce

$$\begin{aligned}\sum_{i=2}^{nx}\sum_{j=2}^{ny}\psi_{i,j}^{T}\delta R_{i,j} &= \cdots+\frac{1}{2}\Big[\psi_{i-1,j}^{T}\left[y_{\eta_{i-\frac{1}{2},j}}A_{i,j}-x_{\eta_{i-\frac{1}{2},j}}B_{i,j}\right]\\ &\quad-\psi_{i+1,j}^{T}\left[y_{\eta_{i+\frac{1}{2},j}}A_{i,j}-x_{\eta_{i+\frac{1}{2},j}}B_{i,j}\right]\\ &\quad+\psi_{i,j}^{T}\Big[\left(y_{\eta_{i+\frac{1}{2},j}}-y_{\eta_{i-\frac{1}{2},j}}-y_{\xi_{i,j+\frac{1}{2}}}+y_{\xi_{i,j-\frac{1}{2}}}\right)A_{i,j}\\ &\qquad-\left(x_{\eta_{i+\frac{1}{2},j}}-x_{\eta_{i-\frac{1}{2},j}}-x_{\xi_{i,j+\frac{1}{2}}}+x_{\xi_{i,j-\frac{1}{2}}}\right)B_{i,j}\Big]\\ &\quad+\psi_{i,j-1}^{T}\left[-y_{\xi_{i,j-\frac{1}{2}}}A_{i,j}+x_{\xi_{i,j-\frac{1}{2}}}B_{i,j}\right]\\ &\quad-\psi_{i,j+1}^{T}\left[-y_{\xi_{i,j+\frac{1}{2}}}A_{i,j}+x_{\xi_{i,j+\frac{1}{2}}}B_{i,j}\right]\Big]\delta w_{i,j}+\cdots \end{aligned}\tag{4.25}$$

Reordering the terms in equation (4.25) leads to the following equation

$$\begin{aligned}\sum_{i=2}^{nx}\sum_{j=2}^{ny}\psi_{i,j}^{T}\delta R_{i,j} &= \cdots-\frac{1}{2}\Big[\left(\psi_{i,j}^{T}-\psi_{i-1,j}^{T}\right)\left(y_{\eta_{i-\frac{1}{2},j}}A_{i,j}-x_{\eta_{i-\frac{1}{2},j}}B_{i,j}\right)\\ &\quad+\left(\psi_{i+1,j}^{T}-\psi_{i,j}^{T}\right)\left(y_{\eta_{i+\frac{1}{2},j}}A_{i,j}-x_{\eta_{i+\frac{1}{2},j}}B_{i,j}\right)\\ &\quad+\left(\psi_{i,j+1}^{T}-\psi_{i,j}^{T}\right)\left(-y_{\xi_{i,j+\frac{1}{2}}}A_{i,j}+x_{\xi_{i,j+\frac{1}{2}}}B_{i,j}\right)\\ &\quad+\left(\psi_{i,j}^{T}-\psi_{i,j-1}^{T}\right)\left(-y_{\xi_{i,j-\frac{1}{2}}}A_{i,j}+x_{\xi_{i,j-\frac{1}{2}}}B_{i,j}\right)\Big]\delta w_{i,j}\\ &\quad+\cdots \end{aligned}$$

Take a transpose of the equation and the adjoint convective flux can then be written

as

$$\begin{aligned}\mathcal{R}(\psi) &= \frac{1}{2}\Big[\Big(y_{\eta_{i-\frac{1}{2},j}}A^T_{i,j} - x_{\eta_{i-\frac{1}{2},j}}B^T_{i,j}\Big)(\psi_{i,j}-\psi_{i-1,j}) \\ &\quad + \Big(y_{\eta_{i+\frac{1}{2},j}}A^T_{i,j} - x_{\eta_{i+\frac{1}{2},j}}B^T_{i,j}\Big)(\psi_{i+1,j}-\psi_{i,j}) \\ &\quad + \Big(-y_{\xi_{i,j+\frac{1}{2}}}A^T_{i,j} + x_{\xi_{i,j+\frac{1}{2}}}B^T_{i,j}\Big)(\psi_{i,j+1}-\psi_{i,j}) \\ &\quad + \Big(-y_{\xi_{i,j-\frac{1}{2}}}A^T_{i,j} + x_{\xi_{i,j-\frac{1}{2}}}B^T_{i,j}\Big)(\psi_{i,j}-\psi_{i,j-1})\Big]. \end{aligned} \tag{4.26}$$

Next, define the flux Jacobian matrices for the total flux across the cell face in the computational domain as

$$\widehat{A^T}_{i+\frac{1}{2},j} = y_{\eta_{i+\frac{1}{2},j}}A^T_{i,j} - x_{\eta_{i+\frac{1}{2},j}}B^T_{i,j} \qquad \text{and} \qquad \widehat{B^T}_{i,j+\frac{1}{2}} = -y_{\xi_{i+\frac{1}{2},j}}A^T_{i,j} + x_{\xi_{i+\frac{1}{2},j}}B^T_{i,j}.$$

Finally, the discrete convective flux can be represented by the following expression

$$\begin{aligned}\mathcal{R}(\psi) &= \frac{1}{2}\Big[\widehat{A^T}_{i-\frac{1}{2},j}(\psi_{i,j}-\psi_{i-1,j}) + \widehat{A^T}_{i+\frac{1}{2},j}(\psi_{i+1,j}-\psi_{i,j}) \\ &\quad + \widehat{B^T}_{i,j+\frac{1}{2}}(\psi_{i,j+1}-\psi_{i,j}) + \widehat{B^T}_{i,j-\frac{1}{2}}(\psi_{i,j}-\psi_{i,j-1})\Big]. \end{aligned} \tag{4.27}$$

Note here that if an average of the metrics evaluated at either flux faces were used in the definition of the flux Jacobian matrices for the total flux across the wall in the computational domain, then $\widehat{A^T}_{i+\frac{1}{2},j}$ would reduce to $\widehat{A^T}_{i,j}$. Equation (4.27) would reduce to the following

$$\mathcal{R}(\psi) = \frac{1}{2}\Big[\widehat{A^T}_{i,j}(\psi_{i+1,j}-\psi_{i-1,j}) + \widehat{B^T}_{i,j}(\psi_{i,j+1}-\psi_{i,j-1})\Big]. \tag{4.28}$$

Equation (4.28) is identical to the discretization of the continuous convective flux gradient defined in equation (4.12). This illustrates that the discretization of the continuous and discrete convective fluxes are similar and only differ in the manner the metrics are calculated in each cell. In summary, the metrics across the cell faces are averaged for each cell in each direction in calculating the continuous adjoint flux, but not in calculating the discrete adjoint flux.

From equation (4.28), in the limit that the mesh width reduces to zero, the discrete

adjoint convective flux can be written as

$$\lim_{\Delta\xi\to 0,\Delta\eta\to 0}\mathcal{R}\left(\psi\right)=\frac{1}{2}\left[\widehat{A^T}_{i,j}\left(\psi_{i+1,j}-\psi_{i-1,j}\right)+\widehat{B^T}_{i,j}\left(\psi_{i,j+1}-\psi_{i,j-1}\right)\right].$$

The second order central difference of the Lagrange Multipliers can then be reduced as

$$\lim_{\Delta\xi\to 0}\left[\frac{\psi_{i+1,j}-\psi_{i-1,j}}{2}\right]=\frac{\partial\psi}{\partial\xi}.$$

Finally, the discrete adjoint convective flux term can be written in continuous form as

$$\lim_{\Delta\xi\to 0,\Delta\eta\to 0}\mathcal{R}\left(\psi\right)=\widehat{A^T}\frac{\partial\psi}{\partial\xi}+\widehat{B^T}\frac{\partial\psi}{\partial\eta}.$$

With a change in notation based on the form used in equation (4.11), the continuous form of the discrete adjoint convective flux can be expressed as

$$\begin{aligned}\mathcal{R}\left(\psi\right) &= \widehat{A^T}\frac{\partial\psi}{\partial\xi}+\widehat{B^T}\frac{\partial\psi}{\partial\eta}\\ &= C_1^T\frac{\partial\psi}{\partial\xi_1}+C_2^T\frac{\partial\psi}{\partial\xi_2}\\ &= C_k^T\frac{\partial\psi}{\partial\xi_k}\end{aligned}$$

The expression above is identical to the continuous adjoint equation defined in equation (4.9).

4.3.2 Contributions from the Viscous Flux

In this sub-section, we derive the discrete viscous adjoint fluxes. The procedure is similar to the derivation of the discrete convective adjoint fluxes illustrated in the previous sub-section; however, the task of producing the viscous counterpart is challenging due to the additional terms in the Navier-Stokes equations.

Since only the viscous fluxes will be considered in this section, the first variation of the total residual for the control volume (i, j) can be simplified to the following equation

$$\delta R(w)_{ij} = -\delta f_{v_{i+\frac{1}{2},j}} + \delta f_{v_{i-\frac{1}{2},j}} - \delta f_{v_{i,j+\frac{1}{2}}} + \delta f_{v_{i,j-\frac{1}{2}}},$$

where the total flux through the $(i + \frac{1}{2}, j)$ flux face can be defined as

$$\delta h_{i+\frac{1}{2},j} = -\delta f_{v_{i+\frac{1}{2},j}}.$$

Next the variation of the viscous flux at the cell face can be computed by taking an average of the fluxes at the cell vertex as explained in section 2.2.1 and shown below

$$\delta f_{v_{i+\frac{1}{2},j}} = \delta \left(y_{\eta_{i+\frac{1}{2},j}} f^*_{v_{i+\frac{1}{2},j}} \right) - \delta \left(x_{\eta_{i+\frac{1}{2},j}} g^*_{v_{i+\frac{1}{2},j}} \right), \tag{4.29}$$

where

$$\begin{aligned} f^*_{v_{i+\frac{1}{2},j}} &= \frac{1}{2} \left(f_{v_{i+\frac{1}{2},j+\frac{1}{2}}} + f_{v_{i+\frac{1}{2},j-\frac{1}{2}}} \right) \\ g^*_{v_{i+\frac{1}{2},j}} &= \frac{1}{2} \left(g_{v_{i+\frac{1}{2},j+\frac{1}{2}}} + g_{v_{i+\frac{1}{2},j-\frac{1}{2}}} \right). \end{aligned}$$

By the chain rule, equation (4.29) can be expanded to produce terms that are multiplied to the variation of the state vector and shape function. Such an expansion would produce the following equation

$$\delta f_{v_{i+\frac{1}{2},j}} = \delta y_{\eta_{i+\frac{1}{2},j}} f^*_{v_{i+\frac{1}{2},j}} + y_{\eta_{i+\frac{1}{2},j}} \delta f^*_{v_{i+\frac{1}{2},j}} - \delta x_{\eta_{i+\frac{1}{2},j}} g^*_{v_{i+\frac{1}{2},j}} - x_{\eta_{i+\frac{1}{2},j}} \delta g^*_{v_{i+\frac{1}{2},j}}. \tag{4.30}$$

We will choose to ignore the metric variations for the rest of the section and concentrate only on expressions that produce terms that are multiplied by the variation

of the state vector. From section 2.2.1 the variation of the viscous flux contribution from the cell vertex $(i+\frac{1}{2}, j+\frac{1}{2})$ can be defined as

$$\delta f_{v_{i+\frac{1}{2},j+\frac{1}{2}}} = \left\{ \begin{array}{c} 0 \\ \delta\tau_{xx} \\ \delta\tau_{xy} \\ \delta u\tau_{xx} + u\delta\tau_{xx} + \delta v\tau_{xy} + v\delta\tau_{xy} + \delta k\frac{\partial T}{\partial x} + k\delta\frac{\partial T}{\partial x} \end{array} \right\}_{i+\frac{1}{2},j+\frac{1}{2}} . \quad (4.31)$$

The variation of the viscous fluxes at the other cell vertexes are defined in a similar fashion. Due to the large number of terms that needs to be considered in the derivation of the discrete viscous adjoint fluxes, the contributions from the momentum and energy equations will be considered in separate sub-sections.

Contributions from the Momentum Equation

In this sub-section, we will concentrate our efforts on the contributions from the momentum equation. First, rewrite equation (4.30) without the variation due to metric terms and substitute f_v^* and g_v^* terms with the average of the viscous fluxes at the cell vertexes,

$$\delta f_{v_{i+\frac{1}{2},j}} = \frac{1}{2} y_{\eta_{i+\frac{1}{2},j}} \delta \left(f_{v_{i+\frac{1}{2},j+\frac{1}{2}}} + f_{v_{i+\frac{1}{2},j-\frac{1}{2}}} \right) - \frac{1}{2} x_{\eta_{i+\frac{1}{2},j}} \delta \left(g_{v_{i+\frac{1}{2},j+\frac{1}{2}}} + g_{v_{i+\frac{1}{2},j-\frac{1}{2}}} \right).$$

Next rearrange the terms to produce the following equation

$$\begin{aligned} \delta f_{v_{i+\frac{1}{2},j}} &= \frac{1}{2} \left[y_{\eta_{i+\frac{1}{2},j}} \delta f_{v_{i+\frac{1}{2},j+\frac{1}{2}}} - x_{\eta_{i+\frac{1}{2},j}} \delta g_{v_{i+\frac{1}{2},j+\frac{1}{2}}} + y_{\eta_{i+\frac{1}{2},j}} \delta f_{v_{i+\frac{1}{2},j-\frac{1}{2}}} - x_{\eta_{i+\frac{1}{2},j}} \delta g_{v_{i+\frac{1}{2},j-\frac{1}{2}}} \right] \\ &= \frac{1}{2} \left[\delta F_{i+\frac{1}{2},j+\frac{1}{2}} + \delta F_{i+\frac{1}{2},j-\frac{1}{2}} \right]. \end{aligned}$$

We now concentrate on the momentum equation in the ξ direction. Then δF can be expressed as

$$\begin{aligned}\delta F_{i+\frac{1}{2},j+\frac{1}{2}} &= y_{\eta_{i+\frac{1}{2},j}}\delta f_{v_{i+\frac{1}{2},j+\frac{1}{2}}} - x_{\eta_{i+\frac{1}{2},j}}\delta g_{v_{i+\frac{1}{2},j+\frac{1}{2}}} \\ &= y_{\eta_{i+\frac{1}{2},j}}\delta \tau_{xx_{i+\frac{1}{2},j+\frac{1}{2}}} - x_{\eta_{i+\frac{1}{2},j}}\delta \tau_{yx_{i+\frac{1}{2},j+\frac{1}{2}}}. \end{aligned} \tag{4.32}$$

The next step is the variation of the stress tensor terms. The purpose of this exercise is to illustrate the procedure and not to show the full derivation; therefore, the expansion will only be carried out for the τ_{xx} term. The viscosity coefficients will be treated as constants in the derivation and therefore its variations are zero and will be neglected in the following derivation,

$$\delta\tau_{xx_{i+\frac{1}{2},j+\frac{1}{2}}} = 2\mu_{i+\frac{1}{2},j+\frac{1}{2}}\delta\left(\frac{\partial u}{\partial x}\right)_{i+\frac{1}{2},j+\frac{1}{2}} + \lambda_{i+\frac{1}{2},j+\frac{1}{2}}\left[\delta\left(\frac{\partial u}{\partial x}\right)_{i+\frac{1}{2},j+\frac{1}{2}} + \delta\left(\frac{\partial v}{\partial y}\right)_{i+\frac{1}{2},j+\frac{1}{2}}\right].$$

Note here that the stress tensor terms are functions of the primitive variables, ρ, u, v, T, and not the state vector, w, which is comprised of $\rho, \rho u, \rho v, \rho E$. In the expansion of the variation of the stress tensor terms, we seek ultimately to produce an equation that is a function of the variation of the primitive variables, $\delta u, \delta v, \ldots$. It will be shown that once the discrete viscous adjoint fluxes are formed, it will be transformed back to a form that is multiplied not by the variation of the primitive variables but by the variation of the state vector. This will allow us to sum the discrete viscous adjoint fluxes to the discrete convective and artificial dissipation fluxes. To simplify the procedure we will attempt to complete the derivation by only collecting terms that are multiplied by the variation of the velocity in the x direction, δu. From section 2.2.1, substitute the velocity gradient terms from equations (2.27) and (2.28) into the above expression for the variation of the stress tensor to produce

$$\begin{aligned}\delta\tau_{xx_{i+\frac{1}{2},j+\frac{1}{2}}} &= \left(\frac{2\mu+\lambda}{2J}\right)_{i+\frac{1}{2},j+\frac{1}{2}}\Big[(y_\eta - y_\xi)_{i+\frac{1}{2},j+\frac{1}{2}}\,\delta u_{i+1,j+1} - (y_\eta + y_\xi)_{i+\frac{1}{2},j+\frac{1}{2}}\,\delta u_{i,j+1} \\ &\qquad + (y_\eta + y_\xi)_{i+\frac{1}{2},j+\frac{1}{2}}\,\delta u_{i+1,j} + (-y_\eta + y_\xi)_{i+\frac{1}{2},j+\frac{1}{2}}\,\delta u_{i,j}\Big],\end{aligned}$$

where J is the cell volume. The variation of the shear stress term, $\delta\tau_{xy}$ can be derived in a similar manner. The variation of the flux at the cell vertex $(i+\frac{1}{2}, j+\frac{1}{2})$ from equation (4.32) can be expressed as

$$\begin{aligned}
\delta F_{i+\frac{1}{2},j+\frac{1}{2}} &= y_{\eta_{i+\frac{1}{2},j}} \left(\frac{2\mu+\lambda}{2J}\right)_{i+\frac{1}{2},j+\frac{1}{2}} \left[(y_\eta - y_\xi)\,\delta u_{i+1,j+1} - (y_\eta + y_\xi)\,\delta u_{i,j+1}\right. \\
&\quad \left. + (y_\eta + y_\xi)\,\delta u_{i+1,j} + (-y_\eta + y_\xi)\,\delta u_{i,j}\right] \\
&\quad - x_{\eta_{i+\frac{1}{2},j}} \left(\frac{\mu}{2J}\right)_{i+\frac{1}{2},j+\frac{1}{2}} \left[(x_\xi - x_\eta)\,\delta u_{i+1,j+1} + (x_\xi + x_\eta)\,\delta u_{i,j+1}\right. \\
&\quad \left. - (x_\xi + x_\eta)\,\delta u_{i+1,j} + (-x_\xi + x_\eta)\,\delta u_{i,j}\right]. \qquad (4.33)
\end{aligned}$$

All metrics terms are evaluated at the $(i+\frac{1}{2}, j+\frac{1}{2})$ vertex. In equation (4.33), the first two lines are contributions from the variation of the normal stress term, $\delta\tau_{xx}$, and the third and fourth lines are contributions from the variation of the shear stress term, $\delta\tau_{yx}$. The expression for $\delta F_{i+\frac{1}{2},j-\frac{1}{2}}$ can be produced by subtracting one from the j subscript from equation (4.33). Thus, the total flux across the $(i+\frac{1}{2}, j)$ flux face can be formulated using the following expression

$$\delta h_{i+\frac{1}{2},j} = -\delta f_{v_{i+\frac{1}{2},j}} = -\frac{1}{2}\left[\delta F_{i+\frac{1}{2},j+\frac{1}{2}} + \delta F_{i+\frac{1}{2},j-\frac{1}{2}}\right].$$

Finally, the variation of the total residual in cell (i,j) can be expressed as

$$\begin{aligned}
\delta R(w)_{i,j} &= -\delta f_{v_{i+\frac{1}{2},j}} + \delta f_{v_{i-\frac{1}{2},j}} - \delta f_{v_{i,j+\frac{1}{2}}} + \delta f_{v_{i,j-\frac{1}{2}}} \\
&= -\frac{1}{2}\left[\delta F_{i+\frac{1}{2},j+\frac{1}{2}} + \delta F_{i+\frac{1}{2},j-\frac{1}{2}}\right] + \frac{1}{2}\left[\delta F_{i-\frac{1}{2},j+\frac{1}{2}} + \delta F_{i-\frac{1}{2},j-\frac{1}{2}}\right] \\
&\quad + \frac{1}{2}\left[\delta G_{i+\frac{1}{2},j+\frac{1}{2}} + \delta G_{i-\frac{1}{2},j+\frac{1}{2}}\right] - \frac{1}{2}\left[\delta G_{i+\frac{1}{2},j-\frac{1}{2}} + \delta G_{i-\frac{1}{2},j-\frac{1}{2}}\right]
\end{aligned}$$

where

$$\begin{aligned}
\delta G_{i+\frac{1}{2},j+\frac{1}{2}} &= -y_{\xi_{i,j+\frac{1}{2}}}\,\delta f_{v_{i+\frac{1}{2},j+\frac{1}{2}}} + x_{\xi_{i,j+\frac{1}{2}}}\,\delta g_{v_{i+\frac{1}{2},j+\frac{1}{2}}} \\
&= -y_{\xi_{i,j+\frac{1}{2}}}\,\delta \tau_{xx_{i+\frac{1}{2},j+\frac{1}{2}}} + x_{\xi_{i,j+\frac{1}{2}}}\,\delta \tau_{yx_{i+\frac{1}{2},j+\frac{1}{2}}}.
\end{aligned}$$

The next step is to pre-multiply the variation of the residual by the transpose of the Lagrange Multiplier, ψ^T, and sum the product over the computational domain to produce

$$\begin{aligned}\sum_{i=2}^{nx}\sum_{j=2}^{ny}\psi_{i,j}^T\delta R_{i,j} &= \cdots+\psi_{i-1,j-1}^T\delta R_{i-1,j-1}+\psi_{i-1,j}^T\delta R_{i-1,j}+\psi_{i-1,j+1}^T\delta R_{i-1,j+1}\\ &\quad+\psi_{i+1,j-1}^T\delta R_{i+1,j-1}+\psi_{i+1,j}^T\delta R_{i+1,j}+\psi_{i+1,j+1}^T\delta R_{i+1,j+1}\\ &\quad+\psi_{i,j-1}^T\delta R_{i,j-1}+\psi_{i,j}^T\delta R_{i,j}+\psi_{i,j+1}^T\delta R_{i,j+1}+\cdots. \qquad (4.34)\end{aligned}$$

The total contribution towards the residual from the viscous fluxes requires information from all eight cells that surrounds cell (i,j). To collect the total contribution towards all the terms that are multiplied by the variation of the u-velocity in the (i,j) cell, $\delta u_{i,j}$, requires the total residual from the surrounding eight cells. Terms that are multiplied by $\delta u_{i,j}$ are collected to produce the total discrete viscous adjoint residual for the second adjoint equation. This is due to the fact that only the $\delta u_{i,j}$ terms are being considered and not the complete viscous flux. After some lengthy algebra the discrete adjoint stress tensor can be expressed as

$$\varpi_{xx_{i+\frac{1}{2},j+\frac{1}{2}}}=2\mu_{i+\frac{1}{2},j+\frac{1}{2}}\left[\frac{\partial\psi_2}{\partial x}\right]_{i+\frac{1}{2},j+\frac{1}{2}}+\lambda_{i+\frac{1}{2},j+\frac{1}{2}}\left\{\left[\frac{\partial\psi_2}{\partial x}\right]+\left[\frac{\partial\psi_3}{\partial y}\right]\right\}_{i+\frac{1}{2},j+\frac{1}{2}}. \qquad (4.35)$$

Note here the remarkable similarity between the Navier-Stokes equation viscous stress tensor expressed in equation (2.26) and discrete viscous adjoint stress tensor shown above. The velocities, u and v, are simply replaced by the second and third adjoint variables. The adjoint variable gradients are expressed as

$$\begin{aligned}\left[\frac{\partial\psi_2}{\partial x}\right]_{i+\frac{1}{2},j+\frac{1}{2}} &= \frac{1}{2J_{i+\frac{1}{2},j+\frac{1}{2}}}\left\{y_{\eta_{i+\frac{1}{2},j+1}}\left[\frac{\partial\psi_2}{\partial\xi}\right]_{i+\frac{1}{2},j+1}+y_{\eta_{i+\frac{1}{2},j}}\left[\frac{\partial\psi_2}{\partial\xi}\right]_{i+\frac{1}{2},j}\right.\\ &\quad\left.-y_{\xi_{i+1,j+\frac{1}{2}}}\left[\frac{\partial\psi_2}{\partial\eta}\right]_{i+1,j+\frac{1}{2}}-y_{\xi_{i,j+\frac{1}{2}}}\left[\frac{\partial\psi_2}{\partial\eta}\right]_{i,j+\frac{1}{2}}\right\}. \qquad (4.36)\end{aligned}$$

Contributions from the Energy Equation

We now focus on the contributions from the energy equation. From equation (4.31) the contribution from the energy equation can be expressed as

$$\delta f_{v_{i+\frac{1}{2},j+\frac{1}{2}}} = \left(\delta u\tau_{xx} + u\delta\tau_{xx} + \delta v\tau_{xy} + v\delta\tau_{xy} + \delta k\frac{\partial T}{\partial x} + k\delta\frac{\partial T}{\partial x}\right)_{i+\frac{1}{2},j+\frac{1}{2}}. \quad (4.37)$$

The contribution can be divided into three parts: variation of the stress tensors, variation of the velocities, and contribution from the heat addition terms,

$$\delta f_{v_{i+\frac{1}{2},j+\frac{1}{2}}} = (u\delta\tau_{xx} + v\delta\tau_{xy})_{i+\frac{1}{2},j+\frac{1}{2}} + (\delta u\tau_{xx} + \delta v\tau_{xy})_{i+\frac{1}{2},j+\frac{1}{2}} + \left(\delta k\frac{\partial T}{\partial x} + k\delta\frac{\partial T}{\partial x}\right)_{i+\frac{1}{2},j+\frac{1}{2}}.$$

First, we consider the contribution of the variation of the stress tensor terms from the energy equation. The variation of the flux at the cell vertex can be written as

$$\begin{aligned}
\delta F_{i+\frac{1}{2},j+\frac{1}{2}} &= y_{\eta_{i+\frac{1}{2},j}}\delta f_{v_{i+\frac{1}{2},j+\frac{1}{2}}} - x_{\eta_{i+\frac{1}{2},j}}\delta g_{v_{i+\frac{1}{2},j+\frac{1}{2}}} \\
&= y_{\eta_{i+\frac{1}{2},j}}\left(u\delta\tau_{xx_{i+\frac{1}{2},j+\frac{1}{2}}} + v\delta\tau_{xy_{i+\frac{1}{2},j+\frac{1}{2}}}\right) \\
&\quad -x_{\eta_{i+\frac{1}{2},j}}\left(u\delta\tau_{yx_{i+\frac{1}{2},j+\frac{1}{2}}} + v\delta\tau_{yy_{i+\frac{1}{2},j+\frac{1}{2}}}\right). \quad (4.38)
\end{aligned}$$

Equation (4.38) is similar to equation (4.32) of the previous section. The main difference between the two equations is the fact that the variation of the stress tensor terms are multiplied by the velocity, u and v. Thus the derivation of the contribution of these terms to the discrete viscous adjoint fluxes follows the derivation of the discrete adjoint stress tensor from the previous section. The contribution of these terms to the discrete viscous adjoint flux can be expressed as

$$\begin{aligned}
\vartheta_{xx_{i+\frac{1}{2},j+\frac{1}{2}}} &= 2\left(u\mu\right)_{i+\frac{1}{2},j+\frac{1}{2}}\left[\frac{\partial\psi_4}{\partial x}\right]_{i+\frac{1}{2},j+\frac{1}{2}} \\
&\quad +\lambda_{i+\frac{1}{2},j+\frac{1}{2}}\left\{\left[u\frac{\partial\psi_4}{\partial x}\right] + \left[v\frac{\partial\psi_4}{\partial y}\right]\right\}_{i+\frac{1}{2},j+\frac{1}{2}}. \quad (4.39)
\end{aligned}$$

Second, consider the contribution of the variation of the velocity components to

the discrete viscous adjoint fluxes. The first step is to express the variation of the flux at the cell vertex with only contributions from terms multiplied by the variation of the velocity components. From equation (4.37), the variation of the flux at the cell vertex can be expressed as

$$\begin{aligned}\delta F_{i+\frac{1}{2},j+\frac{1}{2}} &= y_{\eta_{i+\frac{1}{2},j}}\delta f_{v_{i+\frac{1}{2},j+\frac{1}{2}}} - x_{\eta_{i+\frac{1}{2},j}}\delta g_{v_{i+\frac{1}{2},j+\frac{1}{2}}} \\ &= y_{\eta_{i+\frac{1}{2},j}}\left(\delta u_{i+\frac{1}{2},j+\frac{1}{2}}\tau_{xx_{i+\frac{1}{2},j+\frac{1}{2}}} + \delta v_{i+\frac{1}{2},j+\frac{1}{2}}\tau_{xy_{i+\frac{1}{2},j+\frac{1}{2}}}\right) \\ &\quad -x_{\eta_{i+\frac{1}{2},j}}\left(\delta u_{i+\frac{1}{2},j+\frac{1}{2}}\tau_{yx_{i+\frac{1}{2},j+\frac{1}{2}}} + \delta v_{i+\frac{1}{2},j+\frac{1}{2}}\tau_{yy_{i+\frac{1}{2},j+\frac{1}{2}}}\right). \qquad (4.40)\end{aligned}$$

The flow field velocities are calculated at the cell vertex by averaging the values of the velocities from the four cells that share the same vertex. Concentrating our efforts on the variation of the u velocity component and replacing the equation for the velocity at the cell vertex, equation (4.40) can be simplified to

$$\begin{aligned}\delta F_{i+\frac{1}{2},j+\frac{1}{2}} &= y_{\eta_{i+\frac{1}{2},j}}\delta u_{i+\frac{1}{2},j+\frac{1}{2}}\tau_{xx_{i+\frac{1}{2},j+\frac{1}{2}}} - x_{\eta_{i+\frac{1}{2},j}}\delta u_{i+\frac{1}{2},j+\frac{1}{2}}\tau_{yx_{i+\frac{1}{2},j+\frac{1}{2}}} \\ &= \frac{1}{4}\left(y_{\eta_{i+\frac{1}{2},j}}\tau_{xx_{i+\frac{1}{2},j+\frac{1}{2}}} - x_{\eta_{i+\frac{1}{2},j}}\tau_{yx_{i+\frac{1}{2},j+\frac{1}{2}}}\right)\left[\delta u_{i+1,j+1} + \delta u_{i+1,j} + \delta u_{i,j+1} + \delta u_{i,j}\right].\end{aligned}$$

Since these terms are contributions from the energy equation, they would only be multiplied by the last component of the vector representing the transpose of the Lagrange Multiplier in equation (4.34). After a series of algebraic manipulations, similar to the procedure used from the previous section, the second contribution from the energy equation to the discrete viscous adjoint equation can be written as

$$\varrho_{x_{i+\frac{1}{2},j+\frac{1}{2}}} = \frac{1}{4}\left[\tau_{xx_{i+\frac{1}{2},j+\frac{1}{2}}}\left[\frac{\partial\psi_4}{\partial x}\right]_{i+\frac{1}{2},j+\frac{1}{2}} + \tau_{yx_{i+\frac{1}{2},j+\frac{1}{2}}}\left[\frac{\partial\psi_4}{\partial y}\right]_{i+\frac{1}{2},j+\frac{1}{2}}\right]. \qquad (4.41)$$

The Lagrange Multiplier gradients are defined by equation (4.36). The last contribution from the energy equation to the discrete viscous adjoint flux is from the variation of the heat addition term. From equation (4.37) the variation of the viscous flux can be expressed as

$$\delta f_{v_{i+\frac{1}{2},j+\frac{1}{2}}} = \left[\delta k\frac{\partial T}{\partial x} + k\delta\frac{\partial T}{\partial x}\right]_{i+\frac{1}{2},j+\frac{1}{2}}.$$

If the coefficient of thermal conductivity is treated as a constant, then the only remaining term is the variation of the temperature gradient. The next step is to represent temperature as one of the primitive variables, (ρ, u, v, p). Thus the above expression can be expressed as a function of density and pressure to produce

$$\begin{aligned}\delta f_{v_{i+\frac{1}{2},j+\frac{1}{2}}} &= \left[k\delta\frac{\partial T}{\partial x}\right]_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{1}{\gamma-1}\left[k\delta\frac{\partial}{\partial x}\left(\frac{p}{\rho}\right)\right]_{i+\frac{1}{2},j+\frac{1}{2}} \\ &= \frac{1}{\gamma-1}\left[k\frac{\partial}{\partial x}\left\{\frac{1}{\rho}\delta p - \frac{p}{\rho^2}\delta\rho\right\}\right]_{i+\frac{1}{2},j+\frac{1}{2}}.\end{aligned}$$

After similar algebraic manipulations, the third contribution from the energy equation to the discrete viscous adjoint equation can be written as

$$\varepsilon_{x_{i+\frac{1}{2},j+\frac{1}{2}}} = k_{i+\frac{1}{2},j+\frac{1}{2}}\left[\frac{\partial\psi_4}{\partial x}\right]_{i+\frac{1}{2},j+\frac{1}{2}}. \tag{4.42}$$

Viscous Discrete Adjoint Field Operator

Collecting together the contributions from the momentum and energy equations, the viscous discrete adjoint operator in primitive variables for two-dimensional flow can be expressed as

$$\begin{aligned}(\tilde{L}\psi)_1 &= -\left[\frac{p}{\rho^2}\left(y_\eta - x_\eta\right)\varepsilon_x\right]_{i+\frac{1}{2},j+\frac{1}{2}} + \left[\frac{p}{\rho^2}\left(y_\eta - x_\eta\right)\varepsilon_y\right]_{i+\frac{1}{2},j-\frac{1}{2}} \\ &\quad - \left[\frac{p}{\rho^2}\left(x_\xi - y_\xi\right)\varepsilon_y\right]_{i+\frac{1}{2},j+\frac{1}{2}} + \left[\frac{p}{\rho^2}\left(x_\xi - y_\xi\right)\varepsilon_x\right]_{i-\frac{1}{2},j+\frac{1}{2}} \\ (\tilde{L}\psi)_2 &= \left[y_\eta\left(\varpi_{xx}+\vartheta_{xx}\right) - x_\eta\left(\varpi_{xy}+\vartheta_{xy}\right)\right]_{i+\frac{1}{2},j+\frac{1}{2}} \\ &\quad + \left[y_\eta\left(\varpi_{xx}+\vartheta_{xx}\right) - x_\eta\left(\varpi_{xy}+\vartheta_{xy}\right)\right]_{i+\frac{1}{2},j-\frac{1}{2}} \\ &\quad + \left[x_\xi\left(\varpi_{xy}+\vartheta_{xy}\right) - y_\xi\left(\varpi_{xx}+\vartheta_{xx}\right)\right]_{i+\frac{1}{2},j+\frac{1}{2}} \\ &\quad + \left[x_\xi\left(\varpi_{xy}+\vartheta_{xy}\right) - y_\xi\left(\varpi_{xx}+\vartheta_{xx}\right)\right]_{i-\frac{1}{2},j+\frac{1}{2}} \\ &\quad - \varrho_{x_{i+\frac{1}{2},j+\frac{1}{2}}} - \varrho_{x_{i+\frac{1}{2},j-\frac{1}{2}}} - \varrho_{y_{i+\frac{1}{2},j+\frac{1}{2}}} - \varrho_{y_{i-\frac{1}{2},j+\frac{1}{2}}}\end{aligned}$$

$$
\begin{aligned}
(\tilde{L}\psi)_3 &= \left[y_\eta\left(\varpi_{yx}+\vartheta_{yx}\right)-x_\eta\left(\varpi_{yy}+\vartheta_{yy}\right)\right]_{i+\frac{1}{2},j+\frac{1}{2}} \\
&\quad +\left[y_\eta\left(\varpi_{yx}+\vartheta_{yx}\right)-x_\eta\left(\varpi_{yy}+\vartheta_{yy}\right)\right]_{i+\frac{1}{2},j-\frac{1}{2}} \\
&\quad +\left[x_\xi\left(\varpi_{yy}+\vartheta_{yy}\right)-y_\xi\left(\varpi_{yx}+\vartheta_{yx}\right)\right]_{i+\frac{1}{2},j+\frac{1}{2}} \\
&\quad +\left[x_\xi\left(\varpi_{yy}+\vartheta_{yy}\right)-y_\xi\left(\varpi_{yx}+\vartheta_{yx}\right)\right]_{i-\frac{1}{2},j+\frac{1}{2}} \\
&\quad -\varrho_{y_{i+\frac{1}{2},j+\frac{1}{2}}}-\varrho_{y_{i+\frac{1}{2},j-\frac{1}{2}}}-\varrho_{x_{i+\frac{1}{2},j+\frac{1}{2}}}-\varrho_{x_{i-\frac{1}{2},j+\frac{1}{2}}} \\
(\tilde{L}\psi)_4 &= \left[\frac{1}{\rho}\left(y_\eta-x_\eta\right)\varepsilon_x\right]_{i+\frac{1}{2},j+\frac{1}{2}}-\left[\frac{1}{\rho}\left(y_\eta-x_\eta\right)\varepsilon_y\right]_{i+\frac{1}{2},j-\frac{1}{2}} \\
&\quad +\left[\frac{1}{\rho}\left(x_\xi-y_\xi\right)\varepsilon_y\right]_{i+\frac{1}{2},j+\frac{1}{2}}-\left[\frac{1}{\rho}\left(x_\xi-y_\xi\right)\varepsilon_x\right]_{i-\frac{1}{2},j+\frac{1}{2}}
\end{aligned}
$$

The conservative viscous adjoint operator may now be obtained by the transformation

$$L = M^{-1^T}\widetilde{L}.$$

The transformation matrices M and M^{-1} are provided in appendix A.1.

4.3.3 Contributions from the Artificial Dissipation Flux

In this sub-section we derive the discrete artificial dissipation terms for the adjoint equation. First, let us revisit equation (4.14) that defines the first variation of the total residual for the control volume (i,j)

$$\delta R(w)_{ij} = \delta h_{i+\frac{1}{2},j} - \delta h_{i-\frac{1}{2},j} + \delta h_{i,j+\frac{1}{2}} - \delta h_{i,j-\frac{1}{2}}.$$

If only the artificial dissipation terms are considered, then the total flux at the $(i+\frac{1}{2},j)$ flux face can be defined as

$$\delta h_{i+\frac{1}{2},j} = -\delta d_{i+\frac{1}{2},j}.$$

From equation (2.30), the artificial dissipation scheme used in this work can be written as

$$d_{i+\frac{1}{2},j} = \nu^{(2)}\Lambda_{i+\frac{1}{2},j}(w_{i+1,j} - w_{i,j}) - \nu^{(4)}\Lambda_{i+\frac{1}{2},j}(w_{i+2,j} - 3w_{i+1,j} + 3w_{i,j} - w_{i-1,j}),$$

where $\nu^{(2)}$ and $\nu^{(4)}$ are sensors whose magnitude is scaled proportional to the normalized second difference of the pressure field, Λ is the average spectral radius of the flux Jacobian matrix across the cell face, and w is the state vector. A complete variation of the fluxes would require a variation of every term that is a function of the state vector. Thus a variation of the sensor terms, $\nu^{(2)}$ and $\nu^{(4)}$, requires a variation of the second difference of the pressure field and a variation of the spectral radius would then require a variation of the velocity and the speed of sound term which in itself is a function of the pressure and density fields. This would require an extensive amount of work, and since the magnitude of the dissipative terms is lower than the convective and viscous fluxes, the sensor terms and the spectral radii will be treated as constants in this work and their variation ignored. Accordingly a variation of the artificial dissipation term would result in the following equation,

$$\begin{aligned}
\delta h_{i+\frac{1}{2},j} &= -\delta d_{i+\frac{1}{2},j} \\
\delta h_{i+\frac{1}{2},j} &= -\nu^{(2)}\Lambda_{i+\frac{1}{2},j}(\delta w_{i+1,j} - \delta w_{i,j}) \\
&\quad +\nu^{(4)}\Lambda_{i+\frac{1}{2},j}(\delta w_{i+2,j} - 3\delta w_{i+1,j} + 3\delta w_{i,j} - \delta w_{i-1,j}).
\end{aligned}$$

Next, we examine the variation of the total residual in the control volume. Since we desire only to formulate the dissipation in the ξ direction, then the variation of the total residual in the control volume can be represented as

$$\begin{aligned}
\delta R(w)_{ij} &= \delta h_{i+\frac{1}{2},j} - \delta h_{i-\frac{1}{2},j} \\
&= -\nu^{(2)}\Lambda_{i+\frac{1}{2},j}(\delta w_{i+1,j} - \delta w_{i,j}) \\
&\quad +\nu^{(4)}\Lambda_{i+\frac{1}{2},j}(\delta w_{i+2,j} - 3\delta w_{i+1,j} + 3\delta w_{i,j} - \delta w_{i-1,j}) \\
&\quad +\nu^{(2)}\Lambda_{i-\frac{1}{2},j}(\delta w_{i,j} - \delta w_{i-1,j}) \\
&\quad -\nu^{(4)}\Lambda_{i-\frac{1}{2},j}(\delta w_{i+1,j} - 3\delta w_{i,j} + 3\delta w_{i-1,j} - \delta w_{i-2,j}). \qquad (4.43)
\end{aligned}$$

We must then pre-multiply the variation of the residual by the transpose of the Lagrange Multiplier, sum the product over the computational domain, and isolate terms that are multiplied by the variation of state vector, $\delta w_{i,j}$, in the (i,j) control volume. Since the blended first and third order dissipation scheme used in this work requires a five point stencil, then it is necessary to include the variation of the residual from these five cells. Thus the equation can be represented as

$$\begin{aligned}\sum_{i=2}^{nx}\sum_{j=2}^{ny}\psi_{i,j}^{T}\delta R_{i,j} &= \cdots+\psi_{i-2,j}^{T}\delta R_{i-2,j}+\psi_{i-1,j}^{T}\delta R_{i-1,j}+\psi_{i,j}^{T}\delta R_{i,j} \\ &\quad +\psi_{i+1,j}^{T}\delta R_{i+1,j}+\psi_{i+2,j}^{T}\delta R_{i+2,j}+\cdots . \end{aligned} \tag{4.44}$$

Now substitute the variation of the residual terms from equation (4.43) into equation (4.44). Only terms that are multiplied by the variation of the state vector in the (i,j) cell, $\delta w_{i,j}$, are shown.

$$\begin{aligned}\sum_{i=2}^{nx}\sum_{j=2}^{ny}\psi_{i,j}^{T}\delta R_{i,j} &= \cdots+\psi_{i-2,j}^{T}\nu_{i-\frac{3}{2}.j}^{(4)}\Lambda_{i-\frac{3}{2},j} \\ &\quad -\psi_{i-1,j}^{T}\left[\nu_{i-\frac{1}{2}.j}^{(2)}\Lambda_{i-\frac{1}{2},j}+3\nu_{i-\frac{1}{2}.j}^{(4)}\Lambda_{i-\frac{1}{2},j}+\nu_{i-\frac{3}{2}.j}^{(4)}\Lambda_{i-\frac{3}{2},j}\right] \\ &\quad +\psi_{i,j}^{T}\left[\nu_{i+\frac{1}{2}.j}^{(2)}\Lambda_{i+\frac{1}{2},j}+3\nu_{i+\frac{1}{2}.j}^{(4)}\Lambda_{i+\frac{1}{2},j}+\nu_{i-\frac{1}{2}.j}^{(2)}\Lambda_{i-\frac{1}{2},j}+3\nu_{i-\frac{1}{2}.j}^{(4)}\Lambda_{i-\frac{1}{2},j}\right] \\ &\quad -\psi_{i+1,j}^{T}\left[\nu_{i+\frac{3}{2}.j}^{(4)}\Lambda_{i+\frac{3}{2},j}+\nu_{i+\frac{1}{2}.j}^{(2)}\Lambda_{i+\frac{1}{2},j}+3\nu_{i+\frac{1}{2}.j}^{(4)}\Lambda_{i+\frac{1}{2},j}\right] \\ &\quad +\psi_{i+2,j}^{T}\nu_{i+\frac{3}{2}.j}^{(4)}\Lambda_{i+\frac{3}{2},j}+\cdots . \end{aligned}$$

Then two sets of terms, one for each flux face of the control volume can be formed as follows

$$\begin{aligned}
\sum_{i=2}^{nx}\sum_{j=2}^{ny}\psi_{i,j}^T\delta R_{i,j} &= \cdots + \\
&\quad -\Big[\nu^{(2)}_{i+\frac{1}{2},j}\Lambda_{i+\frac{1}{2},j}\left(\psi^T_{i+1,j}-\psi^T_{i,j}\right)+\nu^{(4)}_{i+\frac{3}{2},j}\Lambda_{i+\frac{3}{2},j}\left(\psi^T_{i+2,j}-\psi^T_{i+1,j}\right) \\
&\qquad -2\nu^{(4)}_{i+\frac{1}{2},j}\Lambda_{i+\frac{1}{2},j}\left(\psi^T_{i+1,j}-\psi^T_{i,j}\right)+\nu^{(4)}_{i-\frac{1}{2},j}\Lambda_{i-\frac{1}{2},j}\left(\psi^T_{i,j}-\psi^T_{i-1,j}\right)\Big] \\
&\quad +\Big[\nu^{(2)}_{i-\frac{1}{2},j}\Lambda_{i-\frac{1}{2},j}\left(\psi^T_{i,j}-\psi^T_{i-1,j}\right)-\nu^{(4)}_{i+\frac{1}{2},j}\Lambda_{i+\frac{1}{2},j}\left(\psi^T_{i+1,j}-\psi^T_{i,j}\right) \\
&\qquad +2\nu^{(4)}_{i-\frac{1}{2},j}\Lambda_{i-\frac{1}{2},j}\left(\psi^T_{i,j}-\psi^T_{i-1,j}\right)-\nu^{(4)}_{i-\frac{3}{2},j}\Lambda_{i-\frac{3}{2},j}\left(\psi^T_{i-1,j}-\psi^T_{i-2,j}\right)\Big] \\
&\quad +\cdots .
\end{aligned}$$

We can now define the discrete adjoint blended first and third order artificial dissipation scheme as

$$\begin{aligned}
\mathcal{D}_{i+\frac{1}{2},j} &= \nu^{(2)}_{i+\frac{1}{2},j}\Lambda_{i+\frac{1}{2},j}\left(\psi^T_{i+1,j}-\psi^T_{i,j}\right)-\nu^{(4)}_{i+\frac{3}{2},j}\Lambda_{i+\frac{3}{2},j}\left(\psi^T_{i+2,j}-\psi^T_{i+1,j}\right) \\
&\quad +2\nu^{(4)}_{i+\frac{1}{2},j}\Lambda_{i+\frac{1}{2},j}\left(\psi^T_{i+1,j}-\psi^T_{i,j}\right)-\nu^{(4)}_{i-\frac{1}{2},j}\Lambda_{i-\frac{1}{2},j}\left(\psi^T_{i,j}-\psi^T_{i-1,j}\right). \quad (4.45)
\end{aligned}$$

If a first order artificial dissipation equation is used, then equation (4.45) would reduce to the term associated with $\nu^{(2)}$. In such a case, the discrete adjoint equations are completely independent of the costate variables in the cells below the wall. However, if we use the blended first and third order equation, these values are required. As shown later, a simple zeroth order extrapolation across the wall produces good results.

On further investigation, the discrete adjoint dissipation flux term can be reduced to continuous form using the following steps. First, represent the sensor and spectral radii as

$$\begin{aligned}
\tau^{(2)}_{i+\frac{1}{2},j} &= \nu^{(2)}_{i+\frac{1}{2},j}\Lambda_{i+\frac{1}{2},j} \\
\tau^{(4)}_{i+\frac{1}{2},j} &= \nu^{(4)}_{i+\frac{1}{2},j}\Lambda_{i+\frac{1}{2},j}.
\end{aligned}$$

As the mesh width is reduced, in the limit that $\Delta\xi \to 0$, the first order difference of

the Lagrange Multiplier can be written as

$$\lim_{\Delta\xi\to 0}\left[\psi^T_{i+1,j}-\psi^T_{i,j}\right]=\left(\frac{\partial\psi^T}{\partial\xi}\right)_{i+\frac{1}{2},j}.$$

The subscript $(i+\frac{1}{2},j)$ in the above equation is not needed in a continuous domain but it is added in to avoid confusion between the different partial derivatives in the next equation. Then equation (4.45) can be rewritten as

$$\begin{aligned}\mathcal{D}_{i+\frac{1}{2},j} &= \tau^{(2)}_{i+\frac{1}{2},j}\left(\frac{\partial\psi^T}{\partial\xi}\right)_{i+\frac{1}{2},j}-\left[\tau^{(4)}_{i+\frac{3}{2},j}\left(\frac{\partial\psi^T}{\partial\xi}\right)_{i+\frac{3}{2},j}\right.\\ &\qquad\left.-2\tau^{(4)}_{i+\frac{1}{2},j}\left(\frac{\partial\psi^T}{\partial\xi}\right)_{i+\frac{1}{2},j}+\tau^{(4)}_{i-\frac{1}{2},j}\left(\frac{\partial\psi^T}{\partial\xi}\right)_{i-\frac{1}{2},j}\right].\end{aligned}\tag{4.46}$$

The first term represents a first order artificial dissipation term. The coefficient $\tau^{(2)}$ acts as a sensor to provide an upwind bias to the second order central adjoint Euler flux to resolve discontinuities without oscillations. The second term, is clearly a third difference term to eliminate the possibility of odd-even point decoupling in the adjoint solution field. In the limit as the mesh width is reduced, equation (4.46) can be further simplified to continuous form to produce

$$\mathcal{D}=\tau^{(2)}\frac{\partial\psi^T}{\partial\xi}-\frac{\partial^2}{\partial\xi^2}\left[\tau^{(4)}\frac{\partial\psi^T}{\partial\xi}\right].\tag{4.47}$$

In order to verify that equation (4.47) is of the correct form, let us reproduce it by including the artificial dissipation scheme in the derivation of the continuous adjoint equation. The variation of the sum of the Euler convective flux (4.3) and the dissipative flux (2.29) in the steady-state limit can be represented as

$$\frac{\partial}{\partial\xi}\delta F-\epsilon^{(2)}\frac{\lambda}{p}\left|\frac{\partial^2 p}{\partial\xi^2}\right|\frac{\partial}{\partial\xi}\delta w+\epsilon^{(4)}\lambda\frac{\partial^3}{\partial\xi^3}\delta w=0.\tag{4.48}$$

Since the dissipation coefficients, spectral radii, and the normalized second difference of the pressure are treated as constants during the derivation of the discrete adjoint

artificial dissipation term, the above equation can be simplified by representing these terms in the following form

$$\begin{aligned}\tau^{(2)} &= \epsilon^{(2)}\frac{\lambda}{p}\left|\frac{\partial^2 p}{\partial\xi^2}\right| \\ \tau^{(4)} &= \epsilon^{(4)}\lambda.\end{aligned}$$

Next we pre-multiply equation (4.48) by the Lagrange Multiplier and integrate over the domain to produce

$$0 = \int_{\mathcal{D}} \psi^T \frac{\partial}{\partial\xi}\left[\delta F - \tau^{(2)}\frac{\partial}{\partial\xi}\delta w + \tau^{(4)}\frac{\partial^3}{\partial\xi^3}\delta w\right].$$

If ψ is differentiable this may be integrated by parts to give

$$0 = \int_{\mathcal{B}} n\psi^T\left[\delta F - \tau^{(2)}\delta w + \tau^{(4)}\frac{\partial^2}{\partial\xi^2}\delta w\right]d\mathcal{B}_\xi - \int_{\mathcal{D}}\frac{\partial\psi^T}{\partial\xi}\left[\delta F - \tau^{(2)}\delta w + \tau^{(4)}\frac{\partial^2}{\partial\xi^2}\delta w\right]d\mathcal{D}_\xi.$$

Redistribute terms and integrate the third difference artificial dissipation scheme a second time to produce

$$\begin{aligned}0 &= \int_{\mathcal{B}} n\psi^T\left[\delta F - \tau^{(2)}\delta w + \tau^{(4)}\frac{\partial^2}{\partial\xi^2}\delta w\right]d\mathcal{B}_\xi - \int_{\mathcal{D}}\frac{\partial\psi^T}{\partial\xi}\delta F d\mathcal{D}_\xi + \int_{\mathcal{D}}\frac{\partial\psi^T}{\partial\xi}\tau^{(2)}\delta w d\mathcal{D}_\xi \\ &\quad - \int_{\mathcal{B}} n\frac{\partial\psi^T}{\partial\xi}\tau^{(4)}\frac{\partial^2}{\partial\xi^2}\delta w d\mathcal{B}_\xi + \int_{\mathcal{D}}\frac{\partial}{\partial\xi}\left[\tau^{(4)}\frac{\partial\psi^T}{\partial\xi}\right]\frac{\partial}{\partial\xi}\delta w d\mathcal{D}_\xi.\end{aligned}$$

A third integration by parts of the last domain integral will produce the final form as

$$\begin{aligned}0 &= \int_{\mathcal{B}} n\psi^T\left[\delta F - \tau^{(2)}\delta w + \tau^{(4)}\frac{\partial^2}{\partial\xi^2}\delta w\right]d\mathcal{B}_\xi - \int_{\mathcal{D}}\frac{\partial\psi^T}{\partial\xi}\delta F d\mathcal{D}_\xi + \int_{\mathcal{D}}\frac{\partial\psi^T}{\partial\xi}\tau^{(2)}\delta w d\mathcal{D}_\xi \\ &\quad - \int_{\mathcal{B}} n\frac{\partial\psi^T}{\partial\xi}\tau^{(4)}\frac{\partial^2}{\partial\xi^2}\delta w d\mathcal{B}_\xi + \int_{\mathcal{B}} n\frac{\partial}{\partial\xi}\left[\tau^{(4)}\frac{\partial\psi^T}{\partial\xi}\right]\frac{\partial}{\partial\xi}\delta w d\mathcal{B}_\xi \\ &\quad - \int_{\mathcal{D}}\frac{\partial^2}{\partial\xi^2}\left[\tau^{(4)}\frac{\partial\psi^T}{\partial\xi}\right]\delta w d\mathcal{D}_\xi.\end{aligned}$$

Since there is no change in the variation of the state vector across the wall, the second and third boundary integrals can be dropped, and the equation can be simplified to

its final form as

$$0 = \int_{\mathcal{B}} n\psi^T \left[\delta F - \tau^{(2)}\delta w\right] d\mathcal{B}_\xi - \int_{\mathcal{D}} \frac{\partial \psi^T}{\partial \xi}\delta F + \left(\tau^{(2)}\frac{\partial \psi^T}{\partial \xi} - \frac{\partial^2}{\partial \xi^2}\left[\tau^{(4)}\frac{\partial \psi^T}{\partial \xi}\right]\right)\delta w d\mathcal{D}_\xi.$$

The expression in the second term of the domain integral is identical to the continuous form of the discrete adjoint artificial dissipation scheme. Therefore the continuous adjoint artificial dissipation flux can be recovered from the discrete adjoint artificial dissipation flux.

4.4 Discrete Adjoint Boundary Conditions

In this section, the viscous adjoint boundary contributions for inverse design and drag minimization are discussed.

4.4.1 Inverse Design

In the case of an inverse design, δI_c is the discrete form of equation (A.15). The $\delta w_{i,2}$ term is added to the corresponding term from equation (4.25), and the metric variation term is added to the gradient term. In contrast with the continuous adjoint, where the boundary condition appears as an update to the Lagrange multipliers in the cell below the wall, the discrete boundary condition appears as a source term in the adjoint fluxes. At cell $(i, 2)$ the adjoint equation is as follows,

$$\begin{aligned} V\frac{\partial \psi_{i,2}}{\partial t} &= \frac{1}{2}\left[-A^T_{i-\frac{1}{2},2}\left(\psi_{i,2} - \psi_{i-1,2}\right) - A^T_{i+\frac{1}{2},2}\left(\psi_{i+1,2} - \psi_{i,2}\right) - B^T_{i,\frac{5}{2}}\left(\psi_{i,3} - \psi_{i,2}\right)\right] \\ &\quad + \mathcal{D}(\psi) + \mathcal{V}(\psi) + \Phi_{inv} \end{aligned} \tag{4.49}$$

where Φ_{inv} is the source term for inverse design,

$$\Phi_{inv} = \left(-\Delta y_\xi \psi_{2_{i,2}} + \Delta x_\xi \psi_{3_{i,2}} - (p - p_T)\Delta s_i\right)\delta p_{i,2}$$

and,

$$A^T_{i+\frac{1}{2},2} = \Delta y_{\eta_{i+\frac{1}{2},2}} \left[\frac{\partial f}{\partial w}\right]^T_{i,2} - \Delta x_{\eta_{i+\frac{1}{2},2}} \left[\frac{\partial g}{\partial w}\right]^T_{i,2}.$$

All the terms in equation (4.49) except for the source term are scaled as the square of Δx. As the mesh width is reduced, the terms within parenthesis in the source term which are divided by Δs_i must approach zero as the solution reaches a steady-state. One then recovers the continuous adjoint boundary condition as stated in equation (A.17).

4.4.2 Drag Minimization

The total drag is a sum of the drag due to pressure and skin friction. In this subsection, the two contributions will be derived separately to form two boundary condition source terms.

Contribution from Pressure Drag

The discrete viscous adjoint boundary condition for pressure drag minimization can be easily obtained by replacing the $(p - p_T)\Delta s_i$ term in the source term of equation (4.49) by the discrete form of equation (A.19). The contribution of the pressure drag cost function to the source term can be expressed as

$$\Phi_{\text{pressure drag}} = \left(-\Delta y_\xi \psi_{2_{i,2}} + \Delta x_\xi \psi_{3_{i,2}} - \frac{2}{\gamma M^2_\infty P_\infty \bar{c}}(\Delta y_\xi \sin\alpha - \Delta x_\xi \cos\alpha)\Delta s_i\right)\delta p_{i,2}$$

Contribution from Skin Friction Drag

Similar to the discrete viscous adjoint boundary conditions for the inverse design case and pressure drag minimization, the discrete viscous adjoint boundary condition for skin friction drag minimization appears as a source term in the adjoint fluxes. At cell

$(i, 2)$ the adjoint source term Φ_v in the direction normal to the surface is as follows,

$$\begin{aligned}\Phi_v = \quad & \mu_{i+\frac{1}{2},j+\frac{1}{2}} \left\{ 2 \left[\frac{\partial \psi_3}{\partial y} \right]_{i+\frac{1}{2},j+\frac{1}{2}} + v_{i+\frac{1}{2},j+\frac{1}{2}} \left[\frac{\partial \psi_4}{\partial y} \right]_{i+\frac{1}{2},j+\frac{1}{2}} - 2 \frac{2}{\gamma M_\infty^2 P_\infty} \Delta x_\xi \sin \alpha \right\} \\ - \quad & \lambda_{i+\frac{1}{2},j+\frac{1}{2}} \left\{ \left[\frac{\partial \psi_2}{\partial x} \right]_{i+\frac{1}{2},j+\frac{1}{2}} + \left[\frac{\partial \psi_3}{\partial y} \right]_{i+\frac{1}{2},j+\frac{1}{2}} + u_{i+\frac{1}{2},j+\frac{1}{2}} \left[\frac{\partial \psi_4}{\partial x} \right]_{i+\frac{1}{2},j+\frac{1}{2}} \right. \\ + \quad & \left. v_{i+\frac{1}{2},j+\frac{1}{2}} \left[\frac{\partial \psi_4}{\partial y} \right]_{i+\frac{1}{2},j+\frac{1}{2}} - \frac{2}{\gamma M_\infty^2 P_\infty} \left(\Delta y_\xi \cos \alpha + \Delta x_\xi \sin \alpha \right) \right\}.\end{aligned}$$

Unlike its counterpart the viscous continuous adjoint boundary condition, the viscous discrete adjoint provides boundary conditions for all four Lagrange multipliers.

The pressure and viscous drag source terms are added separately to the adjoint residual in the cells directly above the wall to satisfy the discrete adjoint boundary condition. In contrast the continuous adjoint formulation does not allow the separation of the pressure and skin friction drag, because the form of the boundary integral in the cost function must be such that it can be canceled by the boundary integral which results from integrating by parts to give equation (4.6).

4.5 Time Integration and Convergence Acceleration

The continuous and discrete adjoint equations are both linear, and consequently they could be solved in one step by direct numerical inversion. The cost of the associated matrix inversion can become prohibitive as the number of mesh cells is increased. Instead, since the equations are similar to the Euler equations, the same iterative method is used to solve both the flow and adjoint equations.

The five stage modified Runge-Kutta time stepping scheme used for the Euler and Navier-Stokes equations described in Chapter 2 is used to march the adjoint equations to the steady-state limit. This greatly simplified the procedure to implement the adjoint module, since both the flow and adjoint solver modules shared the same routines. The local time stepping, implicit residual averaging, and multigrid convergence acceleration procedures used for the flow solver are also applied to the adjoint

module. The smoothing coefficient for the implicit residual averaging are the same as the flow solver. The multigrid implementation is identical to the implementation for the flow solver. Both modules share a large portion of the multigrid subroutines.

4.6 Grid Perturbation

Before we discuss the grid perturbation method used in this work, it is useful to restate the design problem. From equation (4.1), the variation of the cost function written as a function of the variation of the state vector δw, grid point location $\delta \mathcal{X}$, and surface $\delta \mathcal{F}$ can be expressed as

$$\delta I = \frac{\partial I^T}{\partial w}\delta w + \frac{\partial I^T}{\partial \mathcal{X}}\delta \mathcal{X} + \frac{\partial I^T}{\partial \mathcal{F}}\delta \mathcal{F}.$$

The solution of the adjoint equation removes the dependence of the gradient on the flow solution, so that only the variations of the grid point locations and the variation of the surface shape remain.

The variation of the surface shape $\delta \mathcal{F}$ only introduces surface integrals into the equation that computes the gradient. Therefore, the computational cost is negligible even for complex three-dimensional geometries. However, the variations of the grid point locations $\delta \mathcal{X}$ introduce volume integrals into the gradient computation. In order to compute this contribution, regeneration of the grid is required based on perturbations on the surface. The grid regeneration is needed for every surface perturbation. This procedure can be costly if the geometry is three-dimensional and complex, and would have to be repeated a number of times proportional to the number of design variables.

Jameson [35, 38] introduced a grid perturbation method that modifies the current location of the grid points based on perturbations at the geometry surface. The approach is not dependent on the type of structured grid generation used. The method was also successfully used by Burgreen et al. [10]. The method modifies, the grid points along each grid index line projecting from the surface. The arc length between the surface point and the far-field point along the grid line is first computed.

Then the grid points at each location along the grid line is attenuated proportional to its arc length distance from the surface point and the total arc length between the surface and the far-field. The algorithm can be described as

$$\left.\begin{aligned} x_{i,j}^{new} &= x_{i,j}^{old} + C_j \left(x_{i,1}^{new} - x_{i,1}^{old}\right) \\ y_{i,j}^{new} &= y_{i,j}^{old} + C_j \left(y_{i,1}^{new} - y_{i,1}^{old}\right) \end{aligned}\right\} \quad \text{for } i = I,\ j = 2, \ldots, j_{max}, \tag{4.50}$$

where I is the current grid index. The vector C_j can be defined as follows

$$C_j = 1 - (3 - 2\mathcal{N}_j)\,\mathcal{N}_j^2,$$

where $\mathcal{N}$ is the ratio of the arc length from the surface to the current grid point and the total arc length from the surface to the far-field along the grid line as

$$\mathcal{N}_j = \frac{\sum_{l=2}^{j} \sqrt{\left(x_{i,l} - x_{i,l-1}\right)^2 + \left(y_{i,l} - y_{i,l-1}\right)^2}}{\sum_{l=2}^{j_{max}} \sqrt{\left(x_{i,l} - x_{i,l-1}\right)^2 + \left(y_{i,l} - y_{i,l-1}\right)^2}}.$$

From equation (4.50) the variation of the grid point location can be expressed as a function of the variation of the surface points as

$$\delta\mathcal{X} = C_j \delta\mathcal{F}.$$

This allows the variation of the grid point location in the equation for gradient evaluation, to be substituted with the variation of the surface points. The variations of $\delta\mathcal{X}$ and $\delta\mathcal{F}$ are both absorbed into the metric variations δS_{ij} in the derivations of sections 4.2 and 4.3. Reuther [72] provides a complete derivation of the reduction to surface integrals for the Euler equations. This simple grid perturbation scheme has been found to be very robust. Kim [46] verified the robustness of the method by modifying two-dimensional viscous meshes. The grid perturbation method was successful in producing smooth meshes without grid point cross-overs, even in regions of high nonlinearity with large surface perturbations. The grid perturbation method described in this sub-section is ideal for structured meshes, however, the complexity increases

with unstructured meshes. The simplicity in the method is in the effortlessness in producing new grid point locations along the grid line. In the unstructured case, the lack of a continuous grid line extending from the surface to the far-field, removes the efficient property of the grid perturbation method. An alternative, would be not to dampen the grid modification along the grid line but to dampen the changes within a specified bubble around the surface node. The nodes in the unstructured mesh can be shifted based upon their distance from the surface point. Possible alternatives to the grid perturbation scheme have yet to be researched and would be an ideal future work topic.

4.7 Design Variables

The choice of design variables is one of the most crucial steps in any optimization procedure. The success of the optimization of the model problem depends on both the choice of design variables and the cost function. The lift and drag coefficients are common measures of effectiveness of an aerospace vehicle. The system variables that have greatest influence on these parameters are the angle of attack, aircraft configuration, aircraft geometry, etc. In most design processes, the aircraft configuration is generally defined at the end of the conceptual design stage, thus the aircraft configuration can be easily removed as a possible system or design variable, unless the optimal design problem is defined for the conceptual design stage. In this work, the aircraft geometry itself is chosen to be the design variable. In a computational domain, this translates to the surface mesh points. There are several other ways to modify the shape during a design process: such as surface perturbations using the Hicks-Henne "bump" functions [26], B-spline control points, etc. The following two sub-sections describe the implementation used in this work.

4.7.1 Mesh Points

The use of the surface mesh points as design variables ensures that there is no restriction on the attainable geometry. Jameson [35, 38] introduced this concept in his

early work on automatic aerodynamic optimization. If every surface mesh point is used, then the complete design space containing the solution that achieves the global minimum is attainable. In three-dimensional design, this leads to a very large number of design variables. Since the cost of the adjoint approach is independent of the number of design variables, it is feasible to use the surface mesh points as design variables, whereas the cost would be prohibitive if the gradients were computed by the traditional finite-difference method.

Using mesh points as design variables does pose some problems. First, the independent displacement of a single point violates the assumption that the geometry surface is continuous. This would present a difficulty if the flow equation had to be re-evaluated. But since the adjoint approach does not require re-evaluation of the flow, this difficulty is avoided. However, the point-wise gradients may contain high frequency modes, ultimately leading to unsmooth geometry profiles. It is for this reason that Jameson introduced the use of smoothed gradients in the descent procedure, as discussed in section 3.2.2. The smoothed gradient actually corresponds to the use of a weighted Sobolev inner product. Comparison of gradients between the adjoint approach and the finite-difference method are not be possible due to the need for smooth geometry profiles to recompute the flow solutions to obtain finite-difference gradients.

4.7.2 Hicks-Henne Functions

Hicks and Henne [26] parameterized the design space by formulating a set of smooth functions that perturb the geometry surface. The advantage compared to the mesh point approach described in the previous sub-section is the fact that the computed gradient remain smooth and thus no smoothing of the gradient is required. This also ensures that the successive surface geometries in the design process remains smooth. Another advantage is that fewer design variables are required to provide an acceptable span of the design space. Hicks introduced this approach since, the finite-difference method was used to obtain the gradient vector. The Hicks and Henne [26] “sine

bump" functions can be described as

$$b(x) = a\left[\sin\left(\pi x^{\frac{\log 5}{\log t_1}}\right)\right]^{t_2}, \quad \text{for } 0 \leq x \leq 1,$$

where a is the maximum bump magnitude, t_1 locates the maximum point of the bump, and t_2 controls the width of the bump. This flexibility allows one to place the bump at strategic points where a redesign is preferred while leaving other parts of the airfoil intact.

In the main body of this work the mesh points have been used as design variables. However, the Hicks and Henne "sine bump" functions have been used instead of the mesh points when a comparison between the adjoint and finite-difference gradients were desired because it is not feasible to obtain point-wise gradients by the finite-difference method.

4.8 Finite-Difference and Complex-Step Gradients

In order to verify the accuracy of the gradients provided by the adjoint approach, they are compared in section 4.10 to gradients obtained using the finite-difference and complex-step methods. This also provides a confirmation that some of the terms which were dropped in the derivation of the discrete adjoint equations are not important.

Traditionally finite-difference methods have been used to calculate sensitivities of aerodynamic cost functions. The computational cost of the finite-difference method for problems involving large numbers of design variables is both unaffordable and prone to subtractive cancellation error. In order to produce an accurate finite-difference gradient, a range of step sizes must be used, and thus the ultimate cost of producing $\mathcal{N}$ gradient evaluations with the finite-difference method is the product $m\mathcal{N}$, where m is the number of different step sizes used to obtain a converged finite-difference gradient. An estimate of the first derivative of a cost function I using a

first order forward difference approximation is as follows:

$$I^{'}(x) = \frac{I(x+h) - I(x)}{h} + \mathcal{O}(h), \tag{4.51}$$

where h is the step size. A small step size is desired to reduce the truncation error $\mathcal{O}(h)$ but a very small step size would also increase subtractive cancellation errors.

Lyness and Moler [52] introduced the use of the complex-step in calculating the derivative of an analytical function. Here, instead of using a real step h, the step size h is added to the imaginary part of the cost function. A Taylor series expansion of the cost function I yields

$$I(x+ih) = I(x) + ihI^{'}(x) - h^{2}\frac{I^{''}(x)}{2!} - ih^{3}\frac{I^{'''}(x)}{3!} + \cdots .$$

Take the imaginary parts of the above equation and divide by the step size h to produce a second order complex-step approximation to the first derivative:

$$I^{'}(x) = \frac{\mathrm{Im[I(x+}ih)]}{h} + h^{2}\frac{I^{'''}(x)}{3!} + \cdots . \tag{4.52}$$

The complex step formula does not require any subtraction to yield the approximate derivative.

Figure 4.1 illustrates the complex-step versus the finite-difference gradient errors for the inverse design case for decreasing step sizes. At a step size of 10^{-4} the finite-difference and complex-step approximations to the first derivative of the cost function are very similar. As the step size is reduced, the finite-difference gradient error starts to increase because of subtractive cancellation errors; however, the complex-step continues to produce more accurate results. Therefore, the complex-step is more robust and does not require repeated calculations in order to produce an accurate gradient. If a very small step size is chosen, the gradient is calculated only once per design variable. Due to the use of double precision complex numbers, the code requires three times the wall clock time when compared to the finite-difference method. But the benefits of using the complex-step to acquire accurate gradients out-weighs its disadvantages. The code used for this work was modified to handle complex calculations

using an automated method developed by Martins et al. [54, 55].

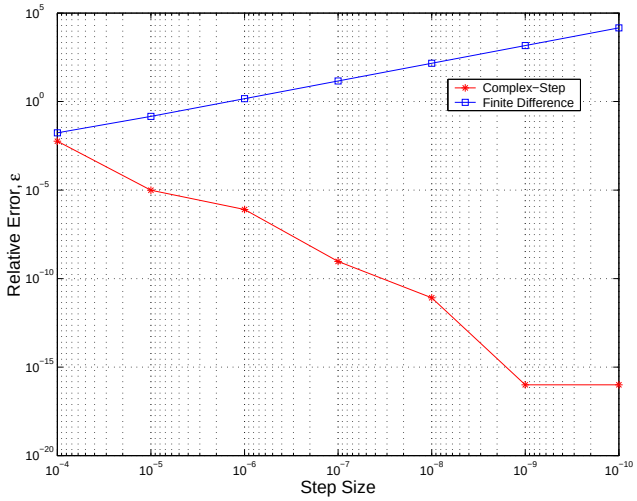


Figure 4.1: Complex-Step Versus Finite-Difference Gradient Errors for Inverse Design Case; $\epsilon = \frac{|g - g_{ref}|}{|g_{ref}|}$

4.9 Outline of the Design Process

The design algorithm can be described as follows:

while ||gradient|| $\leq$ **tolerance**
  Flow Solver.
  Compute Variation of Cost Function.
  Adjoint Solver.
  Compute Gradient.
    for $i = 1, \cdots, N$
      Perturb design variable.
      Modify interior grid points.
      Calculate gradient contribution from each integral.
      Form the gradient vector.
      Implicitly smooth the gradient vector.
    end
  Optimization Algorithm.
  Grid Modification.
    Update surface point based on direction of improvement.
    Impose the thickness constraint.
    Update the interior points.
end while loop

The design procedure listed above is applicable to either the continuous or discrete adjoint approach. The only difference between the two approaches is in the manner the adjoint equation is discretized and the application of the boundary conditions. The boundary conditions are modified based upon the type of cost function. The flow solver module is generally run until at least 5 orders of magnitude drop in the residual is achieved. In the adjoint solver module, the multigrid cycles are repeated until at least 3 orders of magnitude drop in the residual. During the first traverse of the flow and adjoint solvers, approximately two hundred multigrid cycles are employed. During successive design iterations, since the surface modifications are generally small,

only twenty five multigrid cycles are used for the flow and adjoint solvers. The entire process is repeated until the conditions for optimality are satisfied. Figure 4.2 is a graphical representation of the design procedure.

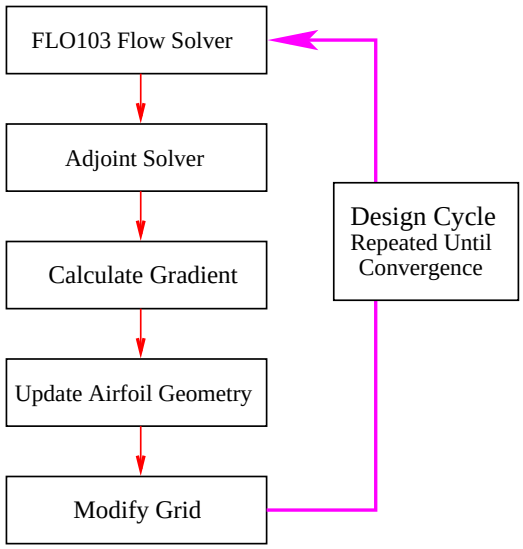


Figure 4.2: Design Procedure

4.10 Results

This section presents the results of the inviscid inverse and drag minimization design, and viscous inverse design and drag minimization cases. For each case, we compare the continuous and discrete adjoint gradients to the finite-difference and complex-step gradients.

4.10.1 Inviscid: Inverse Design

The target pressure is first obtained using the FLO83 flow solver for the NACA 64A410 airfoil at a flight condition of $M_\infty = 0.74$ and a lift coefficient of $C_l = 0.63$ on a 192 x 32 C-grid. At such a condition the NACA 64A410 produces a strong shock

on the upper surface of the airfoil, thus making it an ideal test case for the adjoint versus finite-difference comparison.

The gradient for the continuous and discrete adjoint methods is obtained by perturbing each point on the airfoil. We apply an implicit smoothing technique to the gradient before it is used to obtain a direction of descent for each point on the surface of the airfoil. Figures 4.3 and 4.4 illustrate an inverse design case of a Korn to NACA 64A410 airfoil at fixed lift coefficient. Figure 4.3(a) shows the solution for the Korn airfoil at $M_\infty = 0.74$ and $C_l = 0.63$. After five design cycles we achieve a general shape of the target airfoil as shown in figure 4.3(b). After twenty-five design cycles the upper surface shape is obtained, and nearly eighty percent of the lower surface is achieved. Following a few more iterations, we obtain the desired target pressure except for a few points at the trailing edge. Observe the point-to-point match-up at the shock.

Figures 4.5, 4.6, and 4.7 exhibit the values of the gradients obtained from the adjoint methods and finite-difference for various grid sizes. The circles denote values that we obtain by using the finite difference method. The square represents the discrete adjoint gradient. The asterisk represents the continuous adjoint gradient. The gradient is obtained with respect to variations in Hicks-Henne sine "bump" [26] functions placed along the upper and lower surface of the airfoil. The figures only illustrate the values obtained from the upper surface starting from the leading edge on the left and ending at the trailing edge on the right. In order to reach an accurate finite-difference gradient, we obtain gradients for various step sizes until the finite-difference gradient for each point converges. The discrete adjoint equation is obtained from the discrete flow equations but without taking into account the dependence of the dissipation coefficients on the flow variables. Therefore, in order to eliminate the effect of this on comparisons with the finite-difference gradient we compute the flow solution until attaining a decrease of seven orders of magnitude in the residue. We then freeze the dissipative coefficients and calculate the finite-difference value for each design point. The figures show that the only discrepancies exist in the trailing edge area.

Table 4.1 contains values of the L_2 norm of the difference between the adjoint and

Grid Size	Continuous	Discrete	Cont-Disc
96 x 16	$3.106e-3$	$2.397e-3$	$9.585e-4$
192 x 32	$1.730e-3$	$1.724e-3$	$2.130e-4$
256 x 64	$1.424e-3$	$1.419e-3$	$4.749e-5$

Table 4.1: L_2 norm of the Difference Between Adjoint and Finite-Difference Gradients

finite-difference gradients. The table illustrates three important facts: the difference between the continuous adjoint and finite-difference gradient is slightly greater than that between the discrete adjoint and finite-difference gradient; the norm decreases as the mesh size is increased; and the difference between continuous and discrete adjoint gradients decreases as the mesh size is reduced. The second column depicts the difference between the continuous adjoint and finite-difference gradient. The third column depicts the difference between the discrete adjoint and finite difference gradients. The last column depicts the difference between the discrete adjoint and continuous adjoint. As the mesh size increases the norms decrease as expected. This is due to the fact that the spatial discretization error reduces as the mesh is refined, thus producing a more accurate solution. Since we derive the discrete adjoint by taking a variation of the discrete flow equations, we expect it to be consistent with the finite-difference gradients and thus to be closer than the continuous adjoint to the finite-difference gradient. This is confirmed by numerical results, but the difference is very small. As the mesh size increases, the difference between the continuous and discrete gradients should decrease, and this is reflected in the last column of table 4.1.

Figure 4.8 presents the effect of the partial discretization of the flow solver to obtain the discrete adjoint equation. Here we obtain the finite-difference gradients in the figure without freezing the dissipative coefficients. A small discrepancy exists in regions closer to the leading edge and around the shock. From equation (2.30), the magnitude of the artificial dissipation coefficients for the first and third order blended scheme are proportional to the spectral radii and the normalized second difference of the pressure field. In regions close to high pressure gradients such as the stagnation point which is within the region of the leading edge of the airfoil and discontinuities

such as shock waves, the magnitude of the normalized pressure gradient increases and thus one can no longer assume that the coefficients are constants. The increase in the magnitude of the dissipative terms causes the discrepancies seen in the vicinity of the leading edge and the shock wave of the gradient comparison charts. This study illustrates the importance of a full discretization of the discrete field equations to obtain a complete set of discrete adjoint equations.

Kim et al. [47] have verified that accurate finite-difference gradients require a convergence of four to five orders of magnitude in the flow solver. However, both the continuous and discrete adjoint gradients only require a convergence of two orders of magnitude in the flow solver. Figures 4.9 and 4.10 illustrate the continuous and discrete gradients for various flow solver convergence. In figure 4.11 and 4.12 continuous and discrete adjoint gradients are plotted for various adjoint solver convergence. The gradients only require two orders of magnitude convergence in the adjoint solver. This conclusion illustrates the importance of the control theory approach to automatic aerodynamic optimization. Not only do we save on the cost of computing the gradients, but we also reduce the computational cost, since it is no longer required to converge the flow solution by four or five orders of magnitude to obtain solutions that will provide accurate finite-difference gradients. An automatic design process that computes a large number of initial design options using only flow solutions with two or three orders of magnitude decrease in its residual would allow researchers more time to compute solutions for larger and more complex grids for the final design configurations.

Figure 4.13 shows a comparison of the profiles of the second and third costate values between the continuous and discrete adjoint method in a direction normal to the boundary. The solutions agree in the interior points, differing only at the cell below the boundary due to the different treatment of the boundary condition. In the continuous case the value at cell one is updated by the boundary condition. This is in contrast to the discrete case where the boundary condition appears as a source term when the fluxes are accumulated in cell two and the boundary condition does not depend on the value of the costate in cell one. In figure 4.14 both methods produce similar convergence histories. This result is expected since both the continuous and

discrete adjoint equations use central second order spatial discretizations. In the case of the continuous adjoint approach a central second order spatial discretization was chosen for the convective adjoint flux in order to be consistent with the flow solver which also uses the same spatial discretization. However, for the discrete adjoint equation, the central second order scheme was an outcome of applying control theory to the discrete flow equations. The artificial dissipations schemes are also first and third order differences. Since both approaches use the same time integration method, then the overall convergence rate should be similar.

In figure 4.15 we attempt to design a Korn airfoil based on the target pressure of the NACA 64A410 at a Mach number of 0.78. Both the initial and target pressures contain a very strong shock. A comparison of the finite-difference and adjoint gradients reveals an increase in the discrepancy between the two gradients in the vicinity of the shock. In contrast to figure 4.5, where the shock location is at mesh point 75 along the surface, figure 4.16 illustrates the discrepancy around the stronger shock around mesh point 80.

4.10.2 Inviscid: Drag Minimization

The cost function for drag minimization is the pressure drag of the airfoil. We perform computations on a NACA 64A410 airfoil at a flight condition of $M_\infty = 0.75$ and fixed lift coefficient of $C_l = 0.63$. As before, the gradients are obtained by taking variations respect to Hicks-Henne sine "bump" functions placed along the upper and lower surface of the airfoil. Figure 4.18(a) illustrates the initial solution of the airfoil with 132 drag counts. After two design cycles, the drag is reduced by a third to 44 drag counts. The strong shock in the initial solution is weakened. And after just four design cycles, this value is further halved. In figure 4.19(d), the final design does not contain any shock and the drag count is a mere 15.

Figures 4.20-4.22 illustrate the values of the gradients obtained from the adjoint methods and finite-difference for various grid sizes. The finite-difference gradients are based on the same method used for the inverse design case, where the dissipative coefficients are frozen after a converged flow solution is obtained to simulate a full

Grid Size	Continuous	Discrete	Cont-Disc
96 x 16	$2.920e-2$	$1.275e-2$	$2.009e-2$
192 x 32	$1.049e-2$	$7.577e-3$	$5.072e-3$
256 x 64	$6.241e-3$	$5.542e-3$	$1.315e-3$

Table 4.2: L_2 norm of the Difference Between Adjoint and Finite-Difference Gradients

discretization of the discrete adjoint equation. We reduce the finite-difference step sizes until we gain converged values for each design point. We plot gradients for the upper surface from leading edge to trailing edge. In figure 4.20 design points between 50 and 60 are located in the vicinity of the leading edge, where the gradient has a positive slope. In this region the discrete adjoint gradient agrees better with the finite-difference gradient, if compared to the continuous adjoint gradient. The difference reduces as the grid size increases. The higher curvature in the leading edge compared to the pressure and suction sides of the airfoil leads to larger spatial discretization errors. Therefore an increase to the mesh size increases the number of points that represent the high curvature region in the leading edge and thus reduces the discretization error and this leads to better agreement between the gradients. Apart from the region of the leading edge, the adjoint and finite-difference gradients agree.

Table 4.2 contains values of the L_2 norm of the difference between the adjoint and finite-difference gradients. Similar to the inverse design case, the table illustrates three important facts: the discrete adjoint gradient is closer than the continuous adjoint gradient to the finite-difference gradient; the norms decrease as the mesh size increases; and, finally, the difference between the continuous and discrete adjoint gradient decreases as the mesh size increases.

We recalculate the finite-difference and adjoint gradients in figure 4.23 for the medium size mesh of 192 x 32 cells to illustrate the effect of partial discretization of the flow solver. The dissipative coefficients are not frozen during the finite-difference calculations. A very small discrepancy appears in the leading edge and in the shock wave (points: 137-140).

Figures 4.24 and 4.25 illustrate the continuous and discrete gradients for various

flow solver convergence. Only a single order magnitude drop in the flow solver is required for the adjoint gradients to converge. We plot continuous and discrete adjoint gradients in figure 4.26 and 4.27 for various adjoint solver convergence. The gradients only require one order of magnitude convergence in the adjoint solver.

Figure 4.28 shows a comparison of convergence of the objective function between the continuous and discrete adjoint. Both methods converge to the same value for the objective function. Figure 4.29 presents the second and third costate profiles normal to the boundary for the continuous and discrete adjoint solutions. Both solutions agree in the interior points but disagree at the cell below the wall. This is due to the difference between the enforcement of the boundary condition. Figure 4.30 shows that both adjoint methods produce the same convergence history.

4.10.3 Viscous: Inverse Design

In an inverse design case, the target pressure is generally obtained from a known solution. The target pressure is obtained using the FLO103 flow solver for the NACA0012 airfoil at $M_\infty = 0.75$ and a lift coefficient of $C_l = 0.50$ on a 512x64 C-grid.

Figures 4.31 and 4.32 illustrate an inverse design case of a NACA0012 to Onera M6 airfoil at fixed lift coefficient. Figure 4.31(a) shows the solution for the NACA 0012 airfoil at $M_\infty = 0.75$ and $C_l = 0.50$. After only 4 design cycles, the general shape of the target airfoil is achieved as shown in figure 4.31(b). The circles denote the target pressure distribution, the plus signs are the current upper surface pressure, and lastly, the x marks denote the lower surface pressure distribution. After 100 design iterations the desired target airfoil is obtained. Observe the point-to-point match along the shock. The figures illustrate solutions that are obtained using the continuous adjoint method. The discrete adjoint method produces an identical solution.

Figure 4.33 illustrates another example of an inverse design problem of a RAE to NACA 64A410 airfoil at fixed lift coefficient. Figure 4.33(a) shows the solution for the RAE airfoil at $M_\infty = 0.75$ and $C_l = 0.50$. The final design illustrates that the target airfoil is achieved but with a slight deviation at the shock. The purpose of this example is to illustrate the successful application of the method to unsymmetrical airfoils with

cusp-ed trailing edges. A very strong shock is produced on the upper surface, thus making this an ideal test case for the adjoint versus complex-step gradient comparison.

To ensure that the gradients obtained from the adjoint method are accurate: we first, investigate the sensitivity of the gradient to the convergence level of the flow and adjoint solver; and second, compare them to gradients obtained from a finite-difference or complex-step method. Figure 4.34 shows the adjoint gradient errors for varying flow solver convergence. As seen in the figure, at least a four order magnitude drop in the flow solver convergence is required for adjoint gradients to be accurate up to five significant digits. Any further drop in the flow convergence has a minimal effect on the accuracy of the adjoint gradient; therefore, adjoint gradients as expected are sensitive to the convergence of the flow solver. Note here that an opposite conclusion was reached for the inviscid case, where the adjoint gradients only required the flow solver to converge to only two order magnitude drop in its residual. In the viscous case, the presence of the boundary layer affects the accuracy of the adjoint gradients. However, the gradients are not sensitive to the convergence of the adjoint solver. In figure 4.35(a) one order magnitude drop in the adjoint solver produces gradients that are accurate to four significant digits.

Figure 4.36 illustrates the values of the gradients obtained from the continuous and discrete adjoint and complex-step methods. The asterisks represent the continuous adjoint gradients, the squares represent the discrete adjoint gradients, and the circles denote values that are obtained using the complex-step method. The gradient is obtained with respect to variations in Hicks-Henne sine "bump"[26] functions placed along the upper and lower surfaces of the airfoil. The figure only illustrates the values obtained with modifications to the upper surface starting from the leading edge on the left and ending at the trailing edge on the right. The discrete adjoint equation is obtained from the discrete flow equations but without taking into account the dependence of the dissipation coefficients on the flow variables. Therefore, in order to eliminate the effect of this on comparisons with the complex-step gradient we compute the flow solution until attaining a decrease of five orders of magnitude in the residual. We then freeze the dissipative coefficients and calculate the complex-step value for each design variable.

Grid Size	Continuous	Discrete	Cont-Disc
384 x 64	$1.382e-3$	$1.331e-3$	$8.888e-5$
512 x 64	$1.008e-3$	$9.943e-4$	$4.610e-5$
1024 x 64	$7.809e-4$	$7.795e-4$	$1.425e-5$

Table 4.3: L_2 norm of the Difference Between Adjoint and Complex-Step Gradients

Table 4.3 contains values of the L_2 norm of the difference between the adjoint and complex-step gradients. The table illustrates three important facts: the difference between the discrete adjoint and the complex-step gradient is slightly smaller than that between the continuous adjoint and complex-step gradient; the norm decreases as the mesh size is increased; and the difference between continuous and discrete adjoint gradients decreases as the mesh size is increased. The second column depicts the difference between the continuous adjoint and complex-step gradient, the third column depicts the difference between the discrete adjoint and complex-step gradients, and lastly the last column depicts the difference between the discrete and continuous adjoint. As the mesh size increases, the norm of the difference between adjoint and complex-step decreases as expected. Since we derive the discrete adjoint by taking a variation of the discrete flow equations, we expect it to be consistent with the complex-step gradients and thus to be closer to the complex-step gradient than the continuous adjoint. This is confirmed by numerical results, but the difference is very small. As the mesh size increases, the difference between the continuous and discrete gradients should decrease, and this is reflected in the last column of table 4.3.

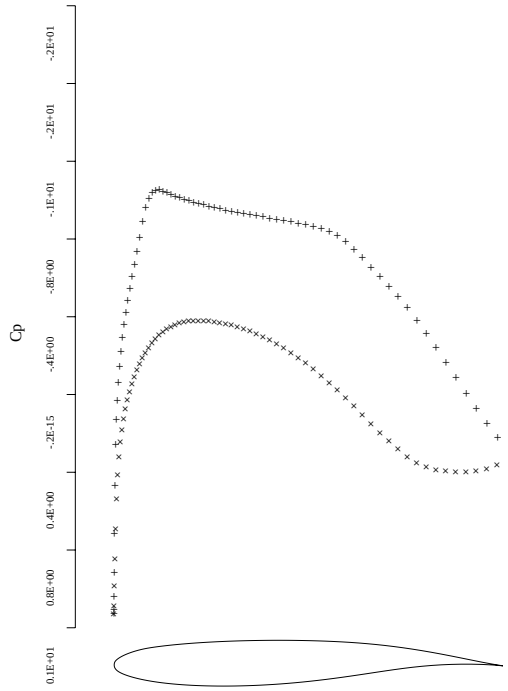
4.10.4 Viscous: Drag Minimization

The drag minimization problem is broken up into two different subsections: pressure drag and total drag minimization. Figures 4.39-4.40 illustrate the drag minimization of RAE 2822 airfoil using the continuous adjoint formulation at a $M_\infty = 0.75$ and a fixed lift coefficient of $C_l = 0.65$. Figure 4.39(a) shows the initial solution of the RAE 2822 airfoil with 56 drag counts due to viscous forces and 92 drag counts due to pressure drag, thus adding up to a total of 148 drag counts. In the first case, as shown in figure 4.39(b), only the pressure drag boundary condition and its

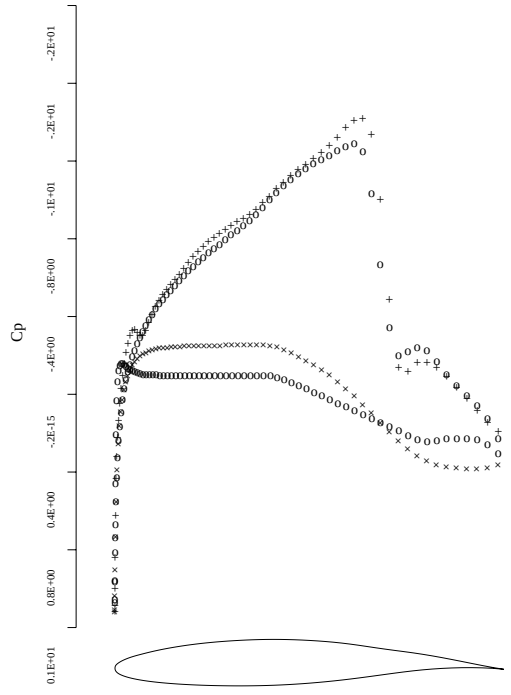
contribution towards the gradient are included. After 20 design iterations, a reduction of 50 drag counts is achieved; however, the skin friction drag increased by 1 count. In figure 4.40(d) the objective function is the total drag coefficient. The resulting airfoil has the same characteristics as the airfoil in figure 4.39(b) that has been obtained by just using the pressure drag.

Figures 4.41 and 4.42 illustrate the pressure and total drag minimization of the RAE 2822 airfoil using the Discrete Adjoint Formulation. In figure 4.41(b) the airfoil has been redesigned by using only the pressure drag boundary condition and its contribution towards the gradient. The solution is similar to the one obtained using the continuous adjoint boundary condition. The pressure drag is reduced by 50 drag counts, but the skin friction drag increases by two drag counts. Thus the total drag reduction is 49 drag counts, compared to the 50 that is obtained with continuous adjoint method. When the total drag minimization boundary condition is used, the discrete adjoint produces the exact same result as the continuous adjoint formulations.

When only the pressure drag boundary condition is used, both the continuous and discrete adjoint gradients match with the complex-step gradient as shown in figure 4.43. Figure 4.44 shows the gradient comparisons for total drag minimization. The discrete adjoint gradient is close to the complex-step gradient, but there are discrepancies between the continuous and complex-step gradients. Kim's results show similar discrepancies, though not as large [46]. The source of these discrepancies is still unknown. This is a topic of future research.

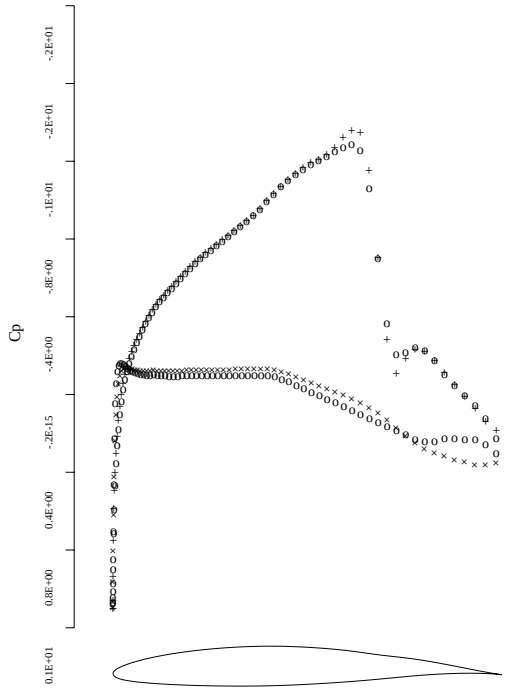


3a: Initial Solution of Korn Airfoil

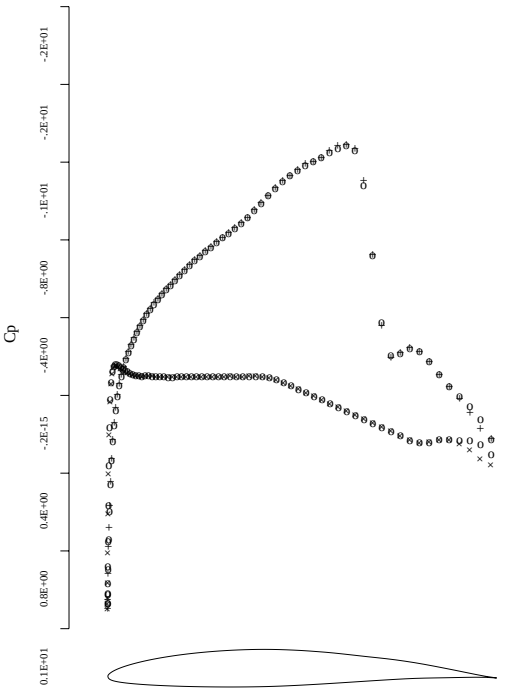


3b: After 5 Design Iterations

Figure 4.3: Inviscid Inverse Design of Korn to NACA 64A410 at Fixed C_l
Grid - 192 x 32, $M = 0.74$, $C_l = 0.63$, $\alpha = 0$ degrees



4a: After 25 Design Iterations



4b: Final Design

Figure 4.4: Inviscid Inverse Design of Korn to NACA 64A410 at Fixed C_l
Grid - 192 x 32, $M = 0.74$, $C_l = 0.63$, $\alpha = 0$ degrees

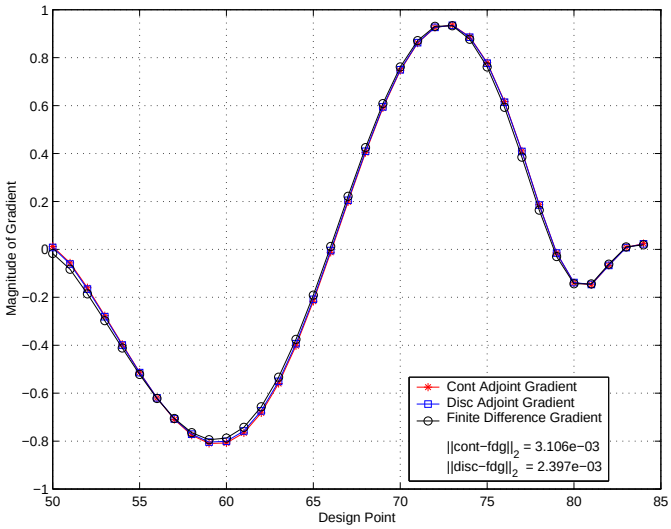


Figure 4.5: Adjoint Versus Finite Difference Gradients for Inviscid Inverse Design of Korn to NACA 64A410 at Fixed C_l. Coarse Grid - 96 x 16, $M = 0.74$, $C_l = 0.63$

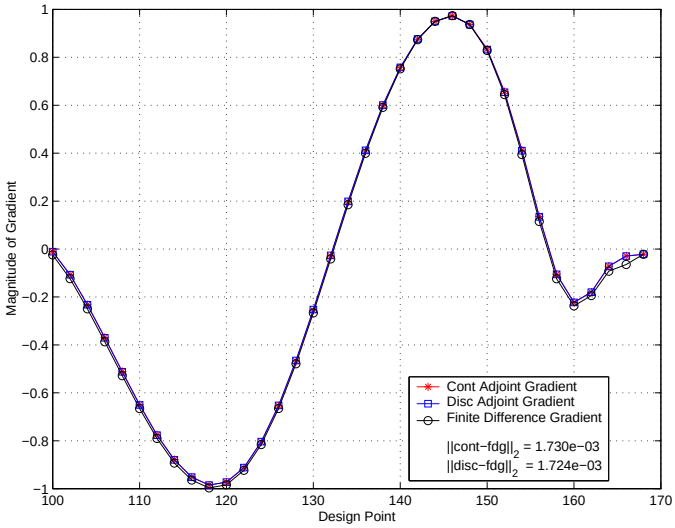


Figure 4.6: Adjoint Versus Finite Difference Gradients for Inviscid Inverse Design of Korn to NACA 64A410 at Fixed C_l. Medium Grid - 192 x 32, $M = 0.74$, $C_l = 0.63$

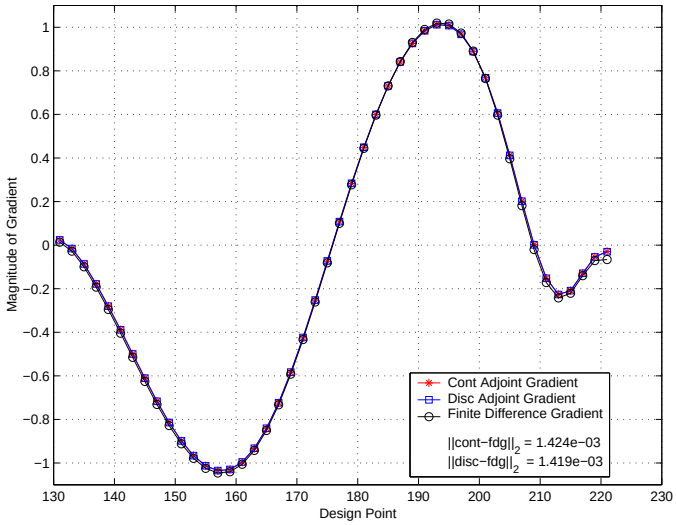


Figure 4.7: Adjoint Versus Finite Difference Gradients for Inviscid Inverse Design of Korn to NACA 64A410 at Fixed C_l. Fine Grid - 256 x 64, $M = 0.74$, $C_l = 0.63$

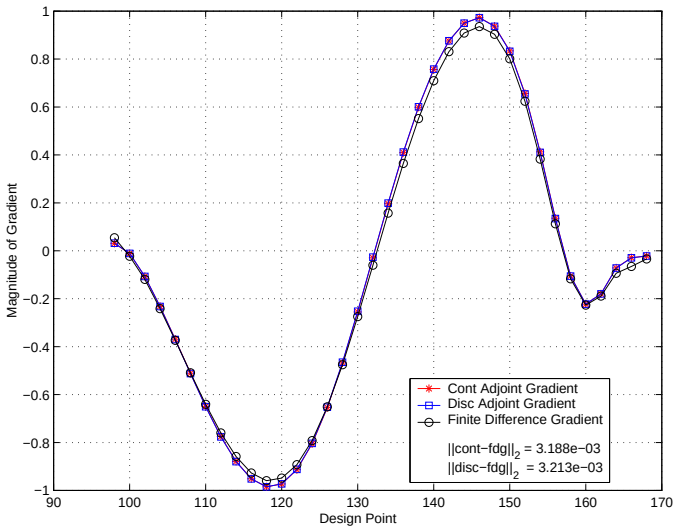


Figure 4.8: Adjoint Versus Finite Difference Gradients for Inviscid Inverse Design of Korn to NACA 64A410 at Fixed C_l. Dissipative Coefficients Not Frozen. Medium Grid - 192 x 32, $M = 0.74$, $C_l = 0.63$

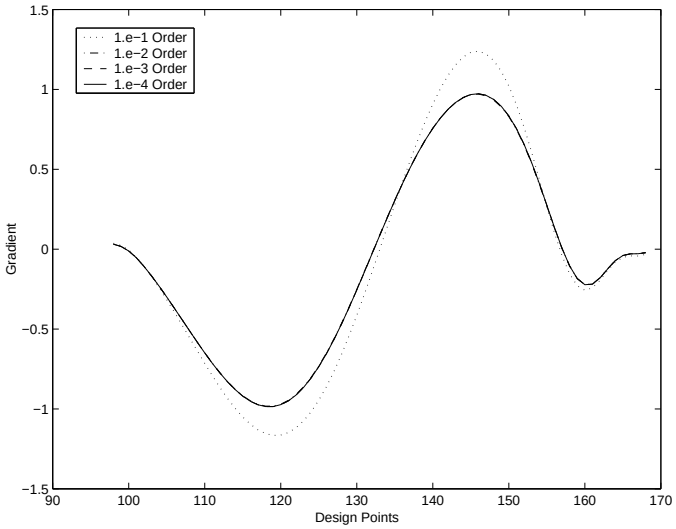


Figure 4.9: Continuous Adjoint Gradients for Varying Flow Solver Convergence for the Inviscid Inverse Design Case

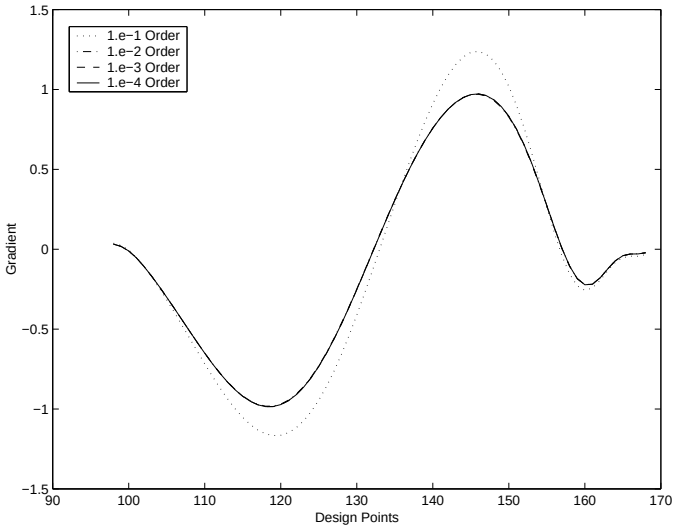


Figure 4.10: Discrete Adjoint Gradients for Varying Flow Solver Convergence for the Inviscid Inverse Design Case

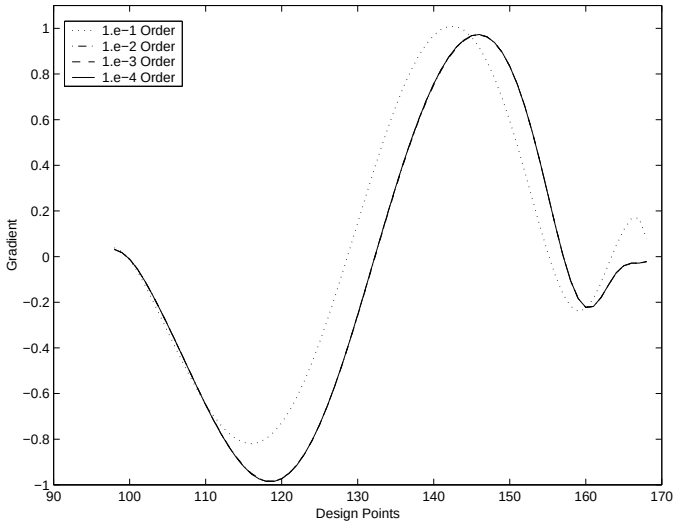


Figure 4.11: Continuous Adjoint Gradients for Varying Adjoint Solver Convergence for the Inviscid Inverse Design Case

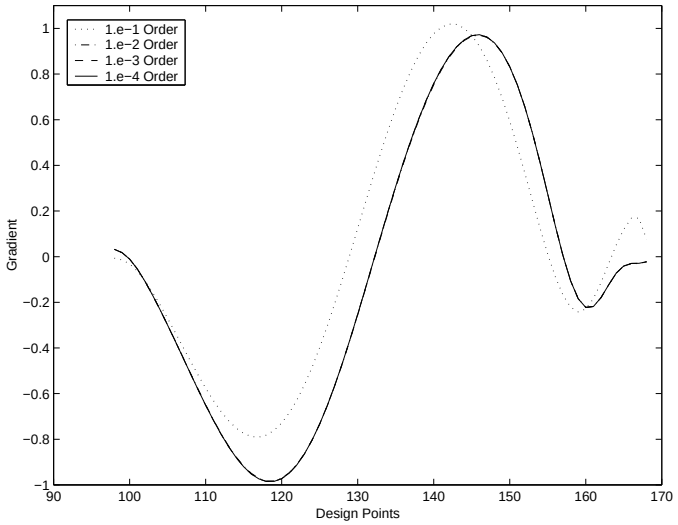


Figure 4.12: Discrete Adjoint Gradients for Varying Adjoint Solver Convergence for the Inviscid Inverse Design Case

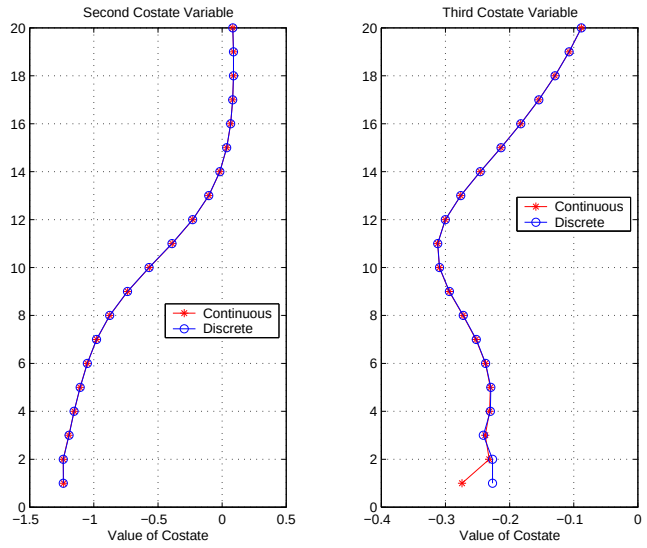


Figure 4.13: Comparison of Costate Values Between the Continuous and Discrete Adjoint Method for the Inviscid Inverse Design of Korn to NACA 64A410 at Fixed C_l. Medium Grid - 192 x 32, $M = 0.74$, $C_l = 0.63$

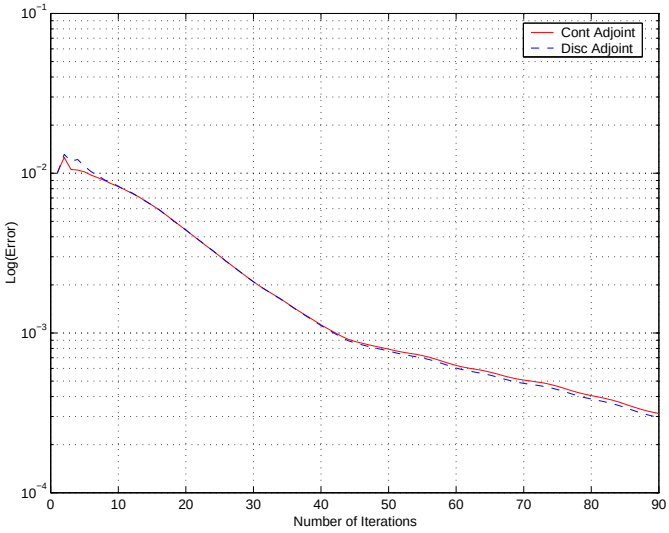
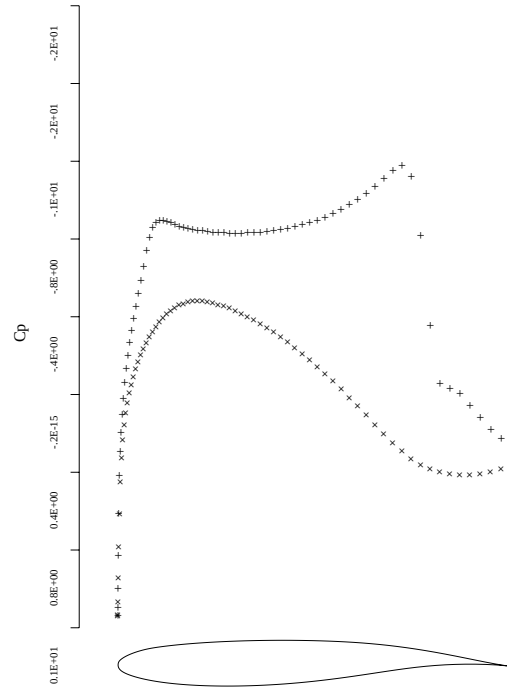
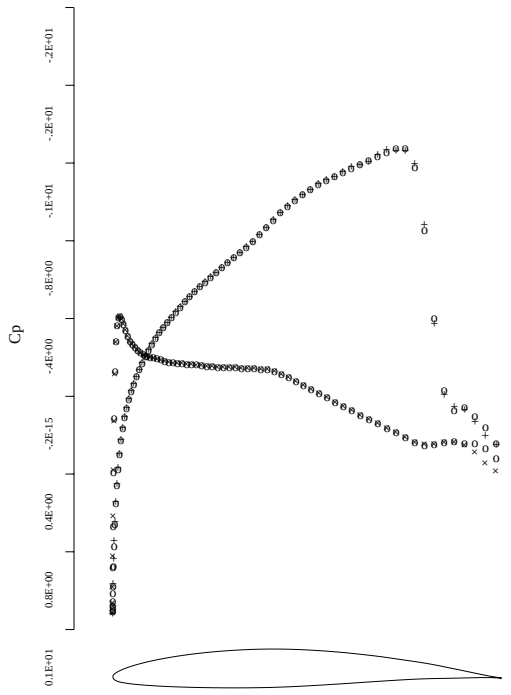


Figure 4.14: Convergence History for the Continuous and Discrete Adjoint for the Inviscid Inverse Design of Korn to NACA 64A410 at Fixed C_l. $M = 0.74$, $C_l = 0.63$



15a: Initial Solution of Korn Airfoil



15b: Final Design

Figure 4.15: Inviscid Inverse Design of Korn to NACA 64A410 at Fixed C_l
Grid - 192 x 32, $M = 0.78$, $C_l = 0.63$, $\alpha = 0$ degrees

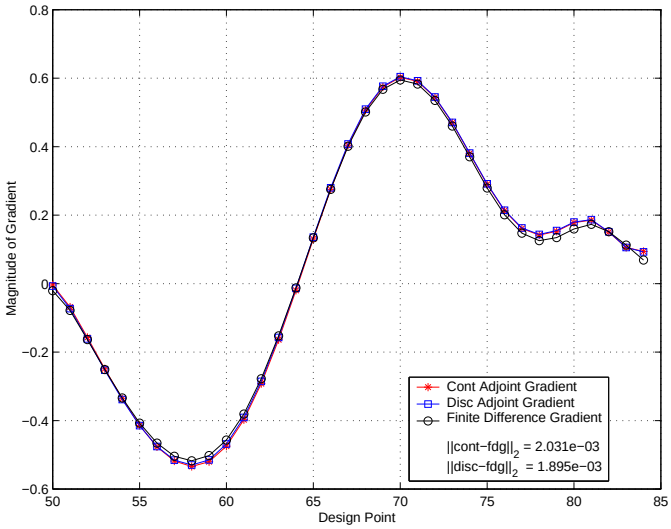


Figure 4.16: Adjoint Versus Finite Difference Gradients for Inviscid Inverse Design of Korn to NACA 64A410 at Fixed C_l. Coarse Grid - 96 x 16, $M = 0.78$, $C_l = 0.63$

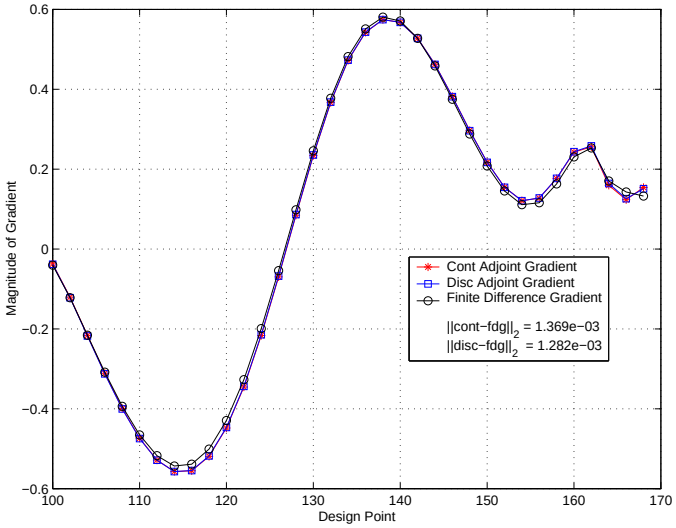
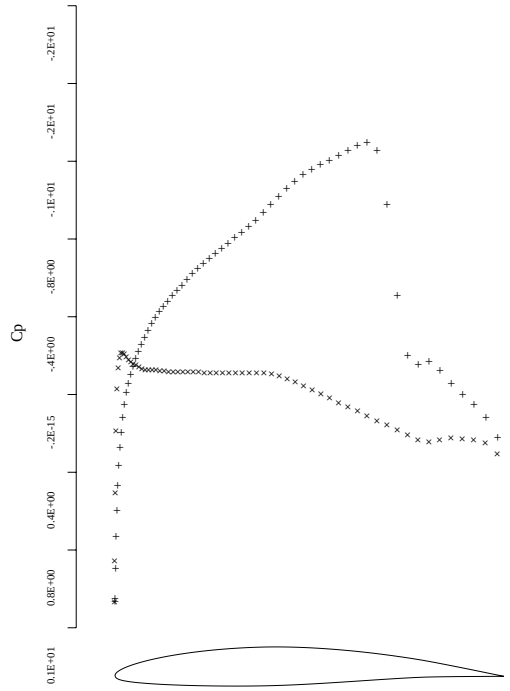
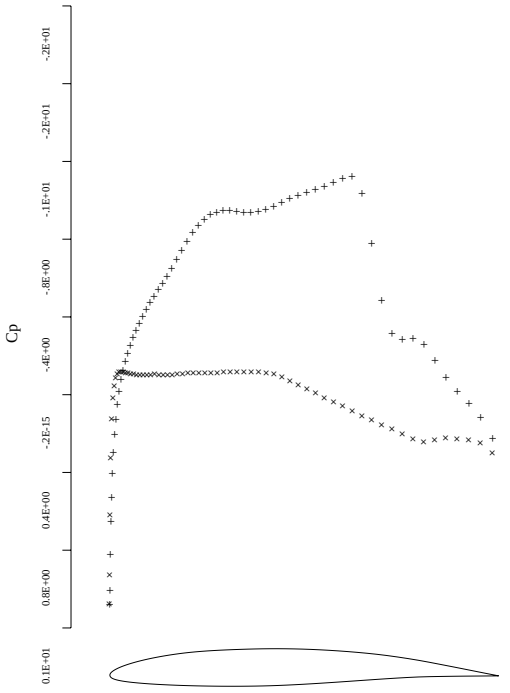


Figure 4.17: Adjoint Versus Finite Difference Gradients for Inviscid Inverse Design of Korn to NACA 64A410 at Fixed C_l. Medium Grid - 192 x 32, $M = 0.78$, $C_l = 0.63$

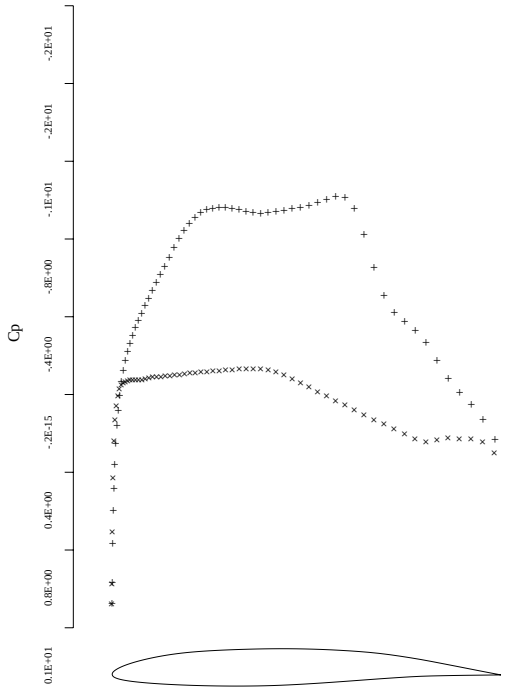


18a: Initial Solution. $C_d = 0.0132$

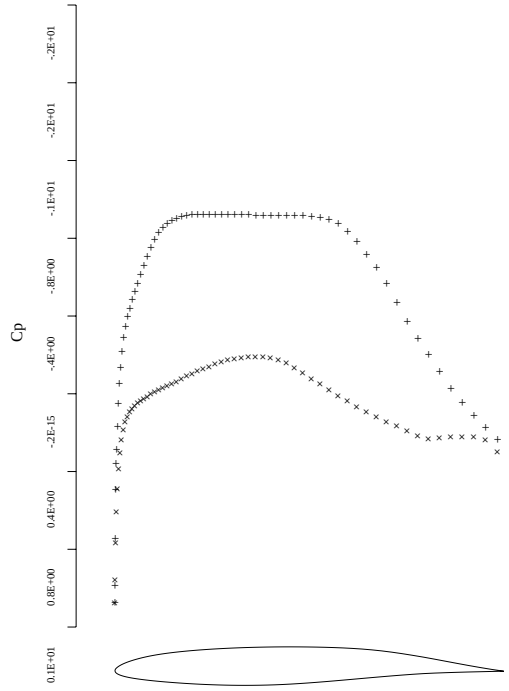


18b: After 2 Design Iterations.
$C_d = 0.0044$

Figure 4.18: Inviscid Drag Minimization of NACA 64A410 at Fixed C_l
Grid - 192 x 32, $M = 0.75$, $C_l = 0.63$



19a: After 4 Design Iterations.
$C_d = 0.0022$



19b: After 20 Design Iterations.
$C_d = 0.0015$

Figure 4.19: Inviscid Drag Minimization of NACA 64A410 at Fixed C_l
Grid - 192 x 32, $M = 0.75$, $C_l = 0.63$

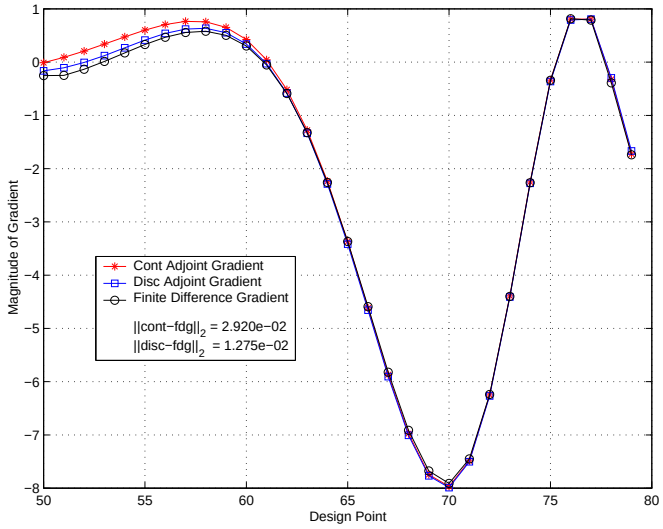


Figure 4.20: Adjoint Versus Finite Difference Gradients for Inviscid Drag Minimization of NACA 64A410 at Fixed C_l.
Coarse Grid - 96 x 16, $M = 0.75$, $C_l = 0.63$

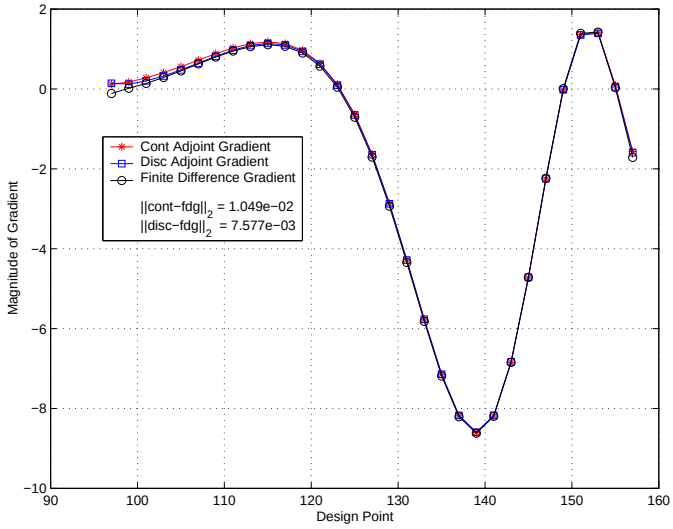


Figure 4.21: Adjoint Versus Finite Difference Gradients for Inviscid Drag Minimization of NACA 64A410 at Fixed C_l.
Medium Grid - 192 x 32, $M = 0.75$, $C_l = 0.63$

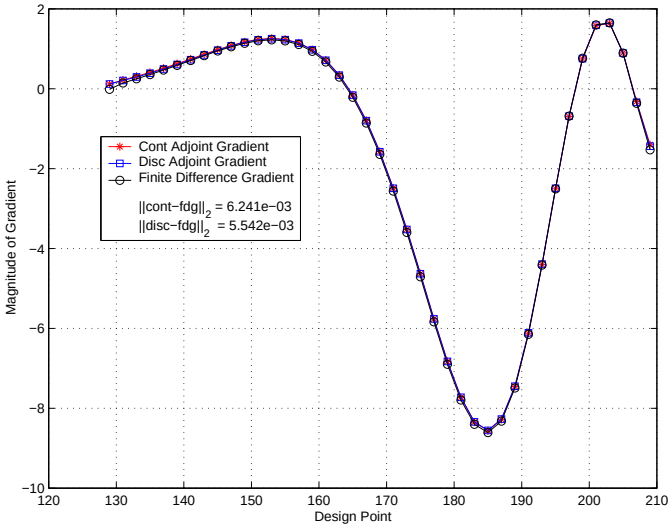


Figure 4.22: Adjoint Versus Finite Difference Gradients for Inviscid Drag Minimization of NACA 64A410 at Fixed C_l.
Fine Grid - 256 x 64, $M = 0.75$, $C_l = 0.63$

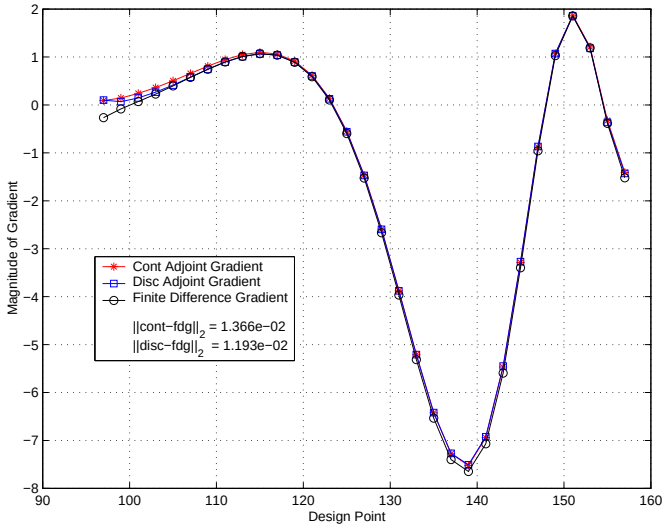


Figure 4.23: Adjoint Versus Finite Difference Gradients for Inviscid Drag Minimization of NACA 64A410 at Fixed C_l. Dissipative Coefficients Not Frozen.
Medium Grid - 192 x 32, $M = 0.75$, $C_l = 0.63$

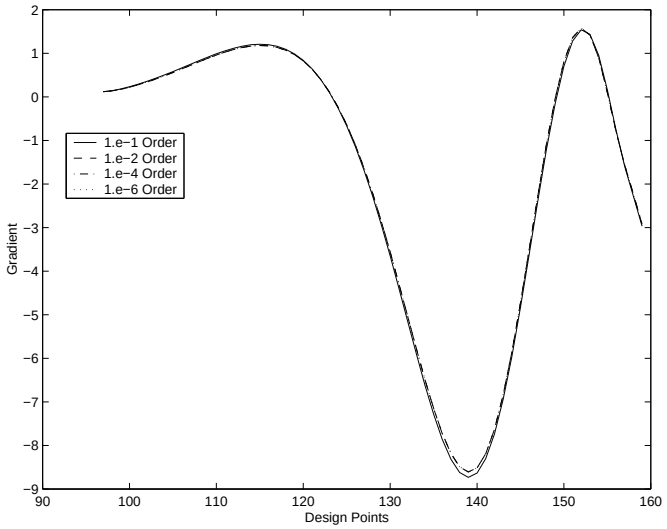


Figure 4.24: Continuous Adjoint Gradients for Varying Flow Solver Convergence for the Inviscid Drag Minimization Case

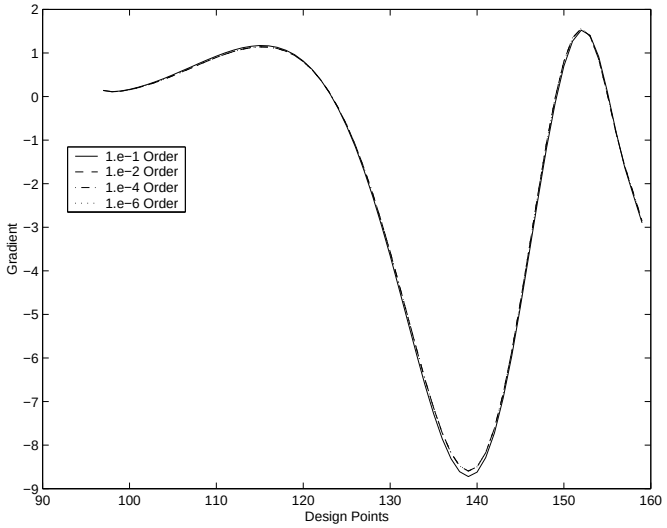


Figure 4.25: Discrete Adjoint Gradients for Varying Flow Solver Convergence for the Inviscid Drag Minimization Case

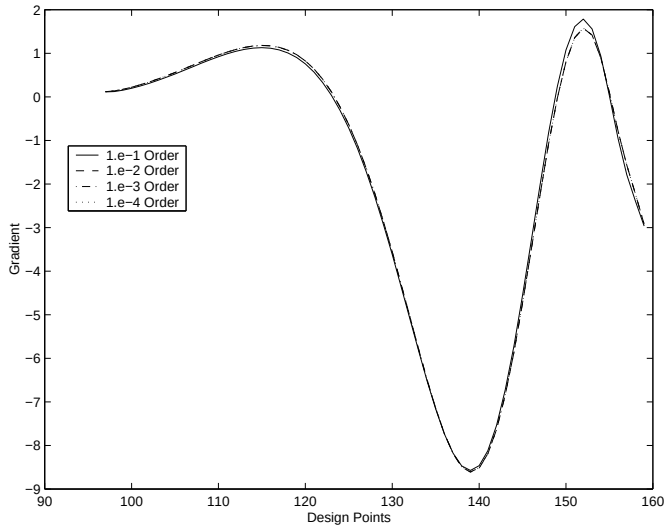


Figure 4.26: Continuous Adjoint Gradients for Varying Adjoint Solver Convergence for the Inviscid Drag Minimization Case

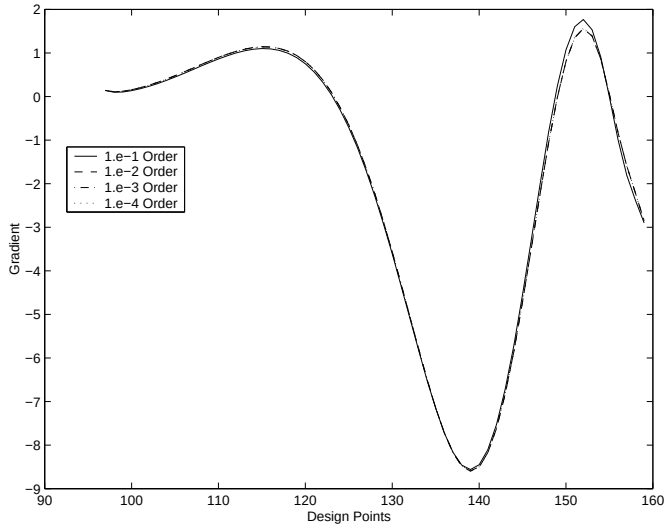


Figure 4.27: Discrete Adjoint Gradients for Varying Adjoint Solver Convergence for the Inviscid Drag Minimization Case

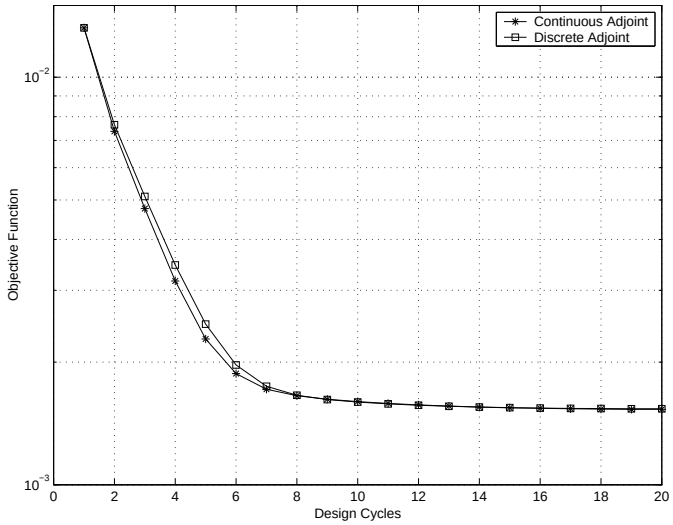


Figure 4.28: Comparison of Convergence of the Objective Function Between the Continuous and Discrete Adjoint Method for Inviscid Drag Minimization.

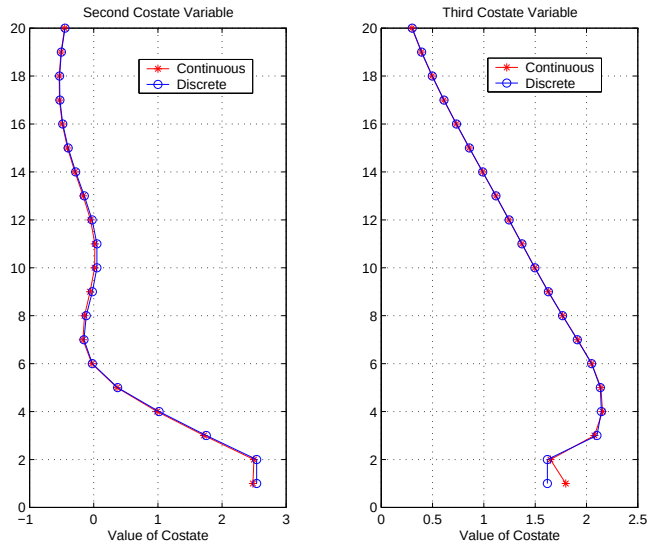


Figure 4.29: Comparison of Costate Values Between the Continuous and Discrete Adjoint Method for Inviscid Drag Minimization of NACA 64A410 at Fixed C_l. Medium Grid - 192 x 32, $M = 0.75$, $C_l = 0.63$

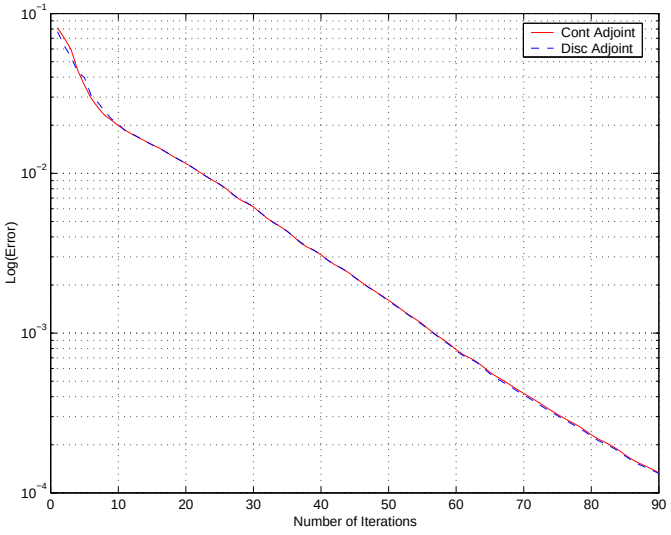
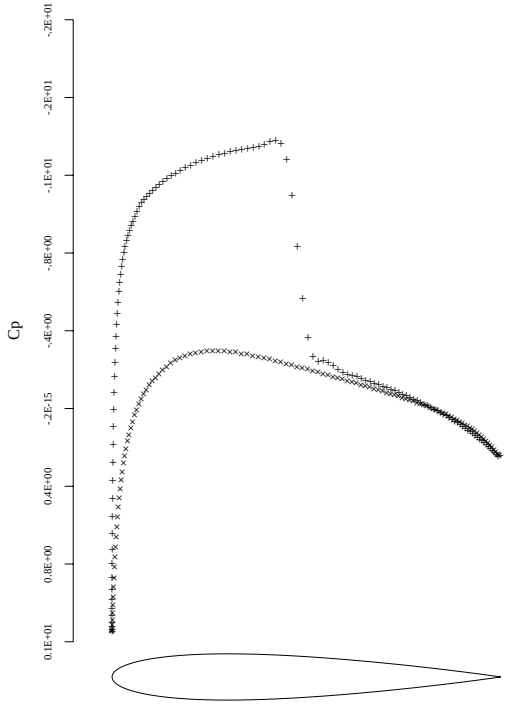
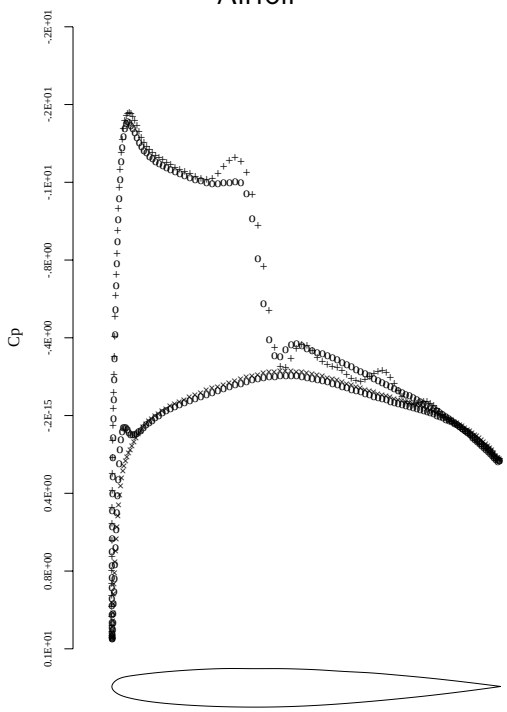


Figure 4.30: Convergence History for the Continuous and Discrete Adjoint for Inviscid Drag Minimization of NACA 64A410 at Fixed C_l. $M = 0.75$, $C_l = 0.63$

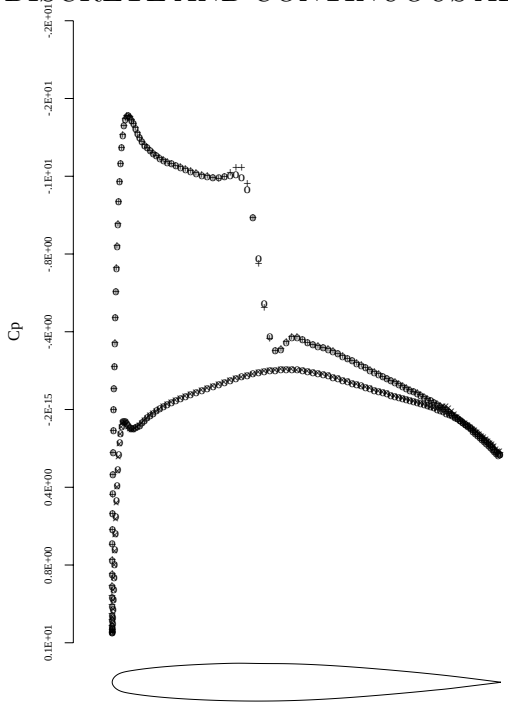


31a: Initial Solution of NACA0012 Airfoil

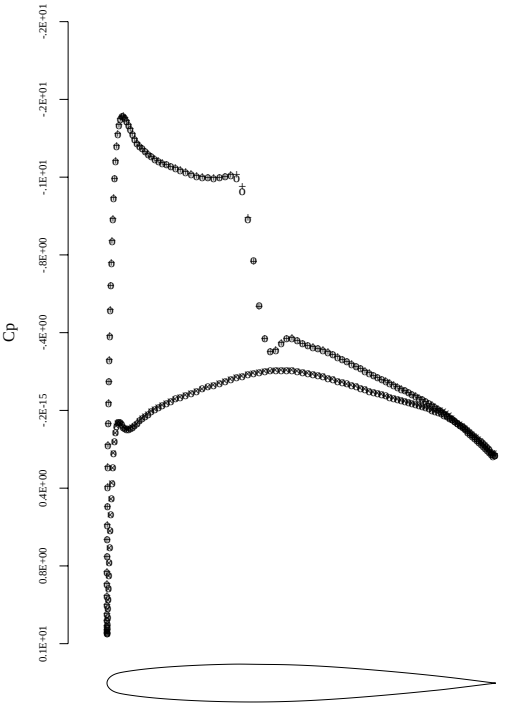


31b: After 4 Design Iterations

Figure 4.31: Inverse Design of NACA 0012 to Onera M6 at Fixed C_l
Grid - 512 x 64, $M = 0.75$, $C_l = 0.65$, $\alpha = 1$ degrees

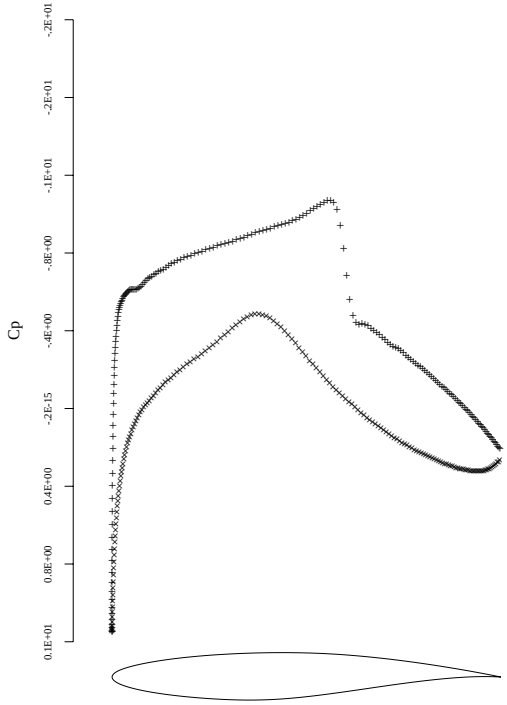


32a: After 50 Design Iterations

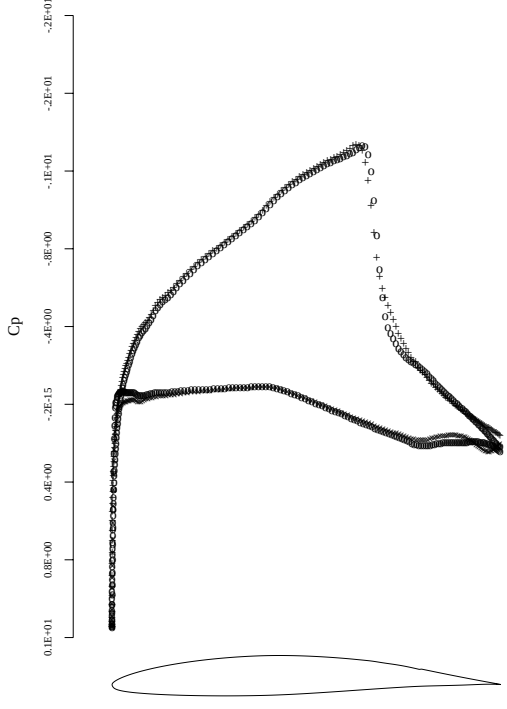


32b: Final Design after 100 Iterations

Figure 4.32: Inverse Design of NACA 0012 to Onera M6 at Fixed C_l
Grid - 512 x 64, $M = 0.75$, $C_l = 0.65$, $\alpha = 1$ degrees



33a: Initial Solution of RAE Airfoil



33b: After 100 Design Iterations

Figure 4.33: Inverse Design of RAE to NACA64A410 at Fixed C_l
Grid - 512 x 64, $M = 0.75$, $C_l = 0.50$, $\alpha = 1$ degrees

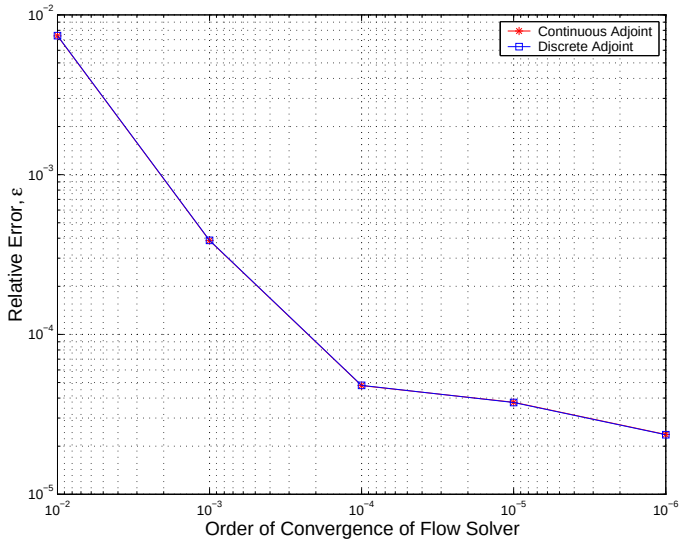


Figure 4.34: Adjoint Gradient Errors for Varying Flow Solver Convergence for the Inverse Design Case; $\epsilon = \frac{|g-g_{ref}|}{|g_{ref}|}$. Fine Grid - 512 x 64, $M = 0.75$, $C_l = 0.65$

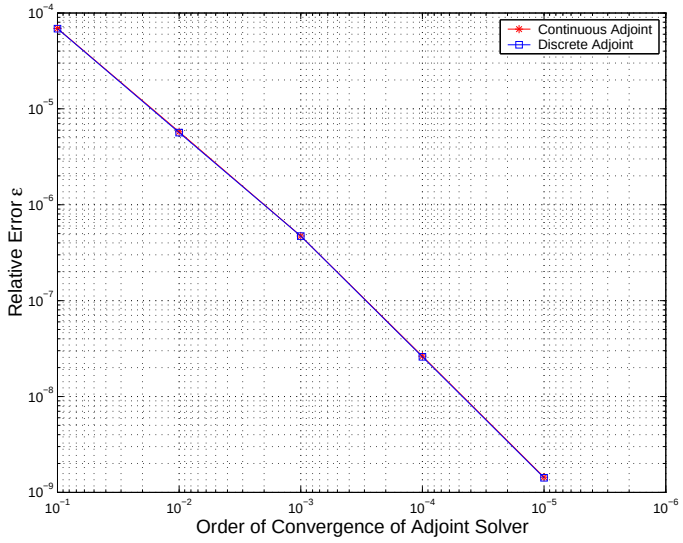


Figure 4.35: Adjoint Gradient Errors for Varying Adjoint Solver Convergence for the Inverse Design Case; $\epsilon = \frac{|g-g_{ref}|}{|g_{ref}|}$. Fine Grid - 512 x 64, $M = 0.75$, $C_l = 0.65$

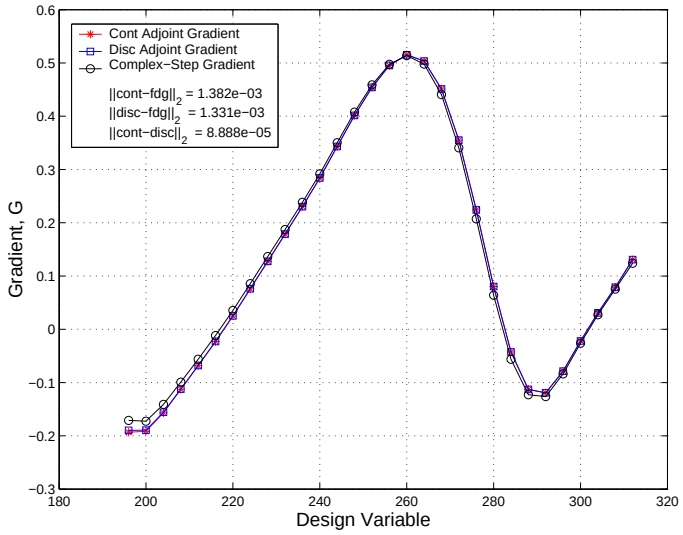


Figure 4.36: Adjoint Versus Complex-Step Gradients for Inverse Design of RAE to NACA64A410 at Fixed C_l. Coarse Grid - 384 x 64, $M = 0.75$, $C_l = 0.65$

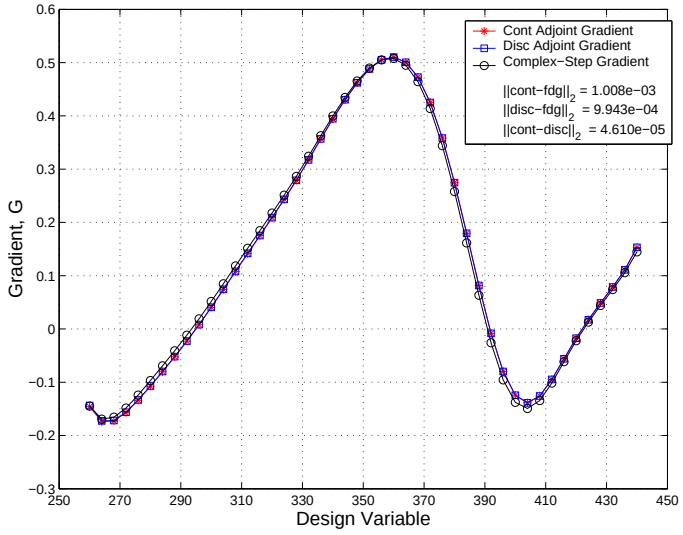


Figure 4.37: Adjoint Versus Complex-Step Gradients for Inverse Design of RAE to NACA64A410 at Fixed C_l. Medium Grid - 512 x 64, $M = 0.75$, $C_l = 0.65$

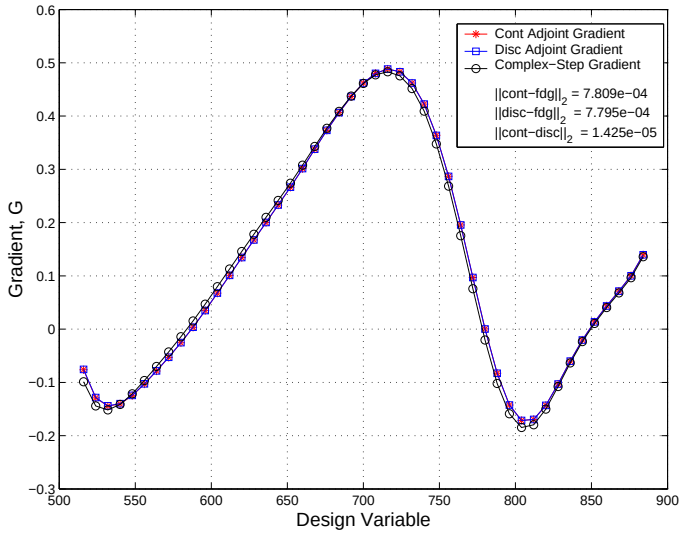
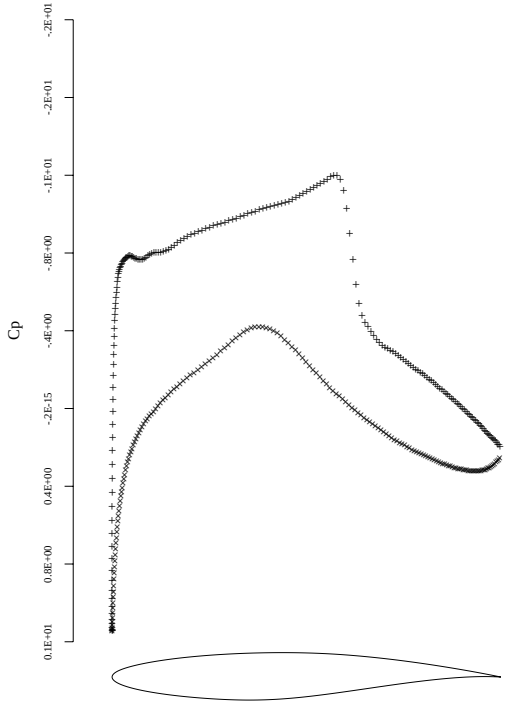
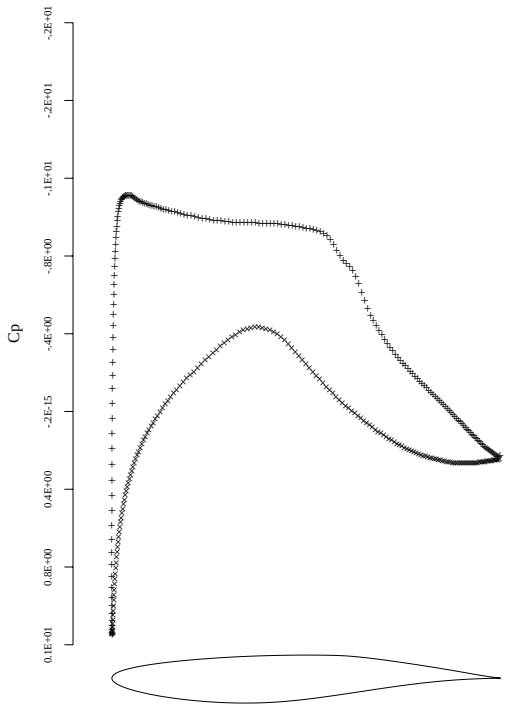


Figure 4.38: Adjoint Versus Complex-Step Gradients for Inverse Design of RAE to NACA64A410 at Fixed C_l. Fine Grid - 1024 x 64, $M = 0.75$, $C_l = 0.65$

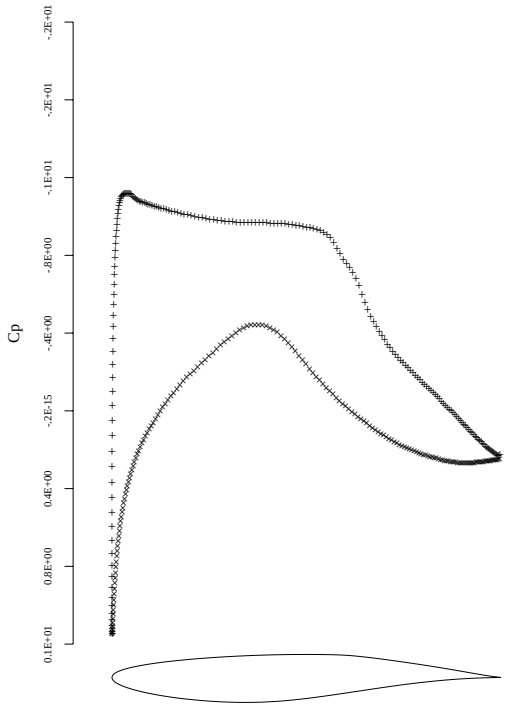


39a: Initial Solution of RAE Airfoil



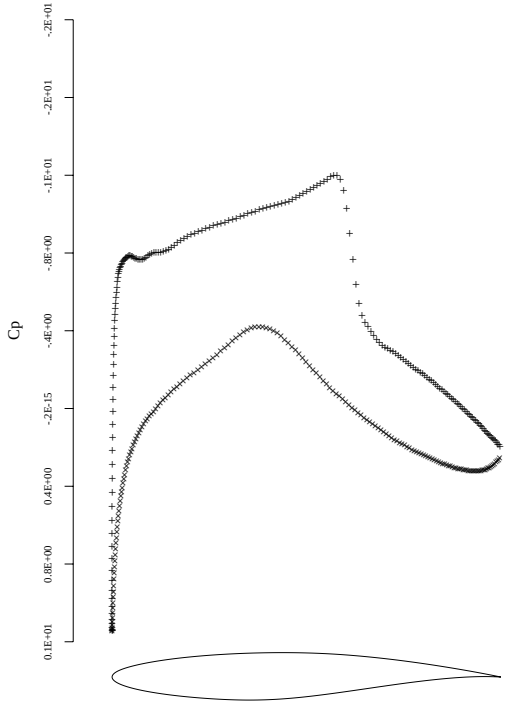
39b: Final Design Based on Pressure Drag Minimization

Figure 4.39: Drag Minimization of RAE Airfoil using the Continuous Adjoint Formulation. Grid - 512 x 64, $M = 0.75$, Fixed $C_l = 0.65$, $\alpha = 1$ degrees

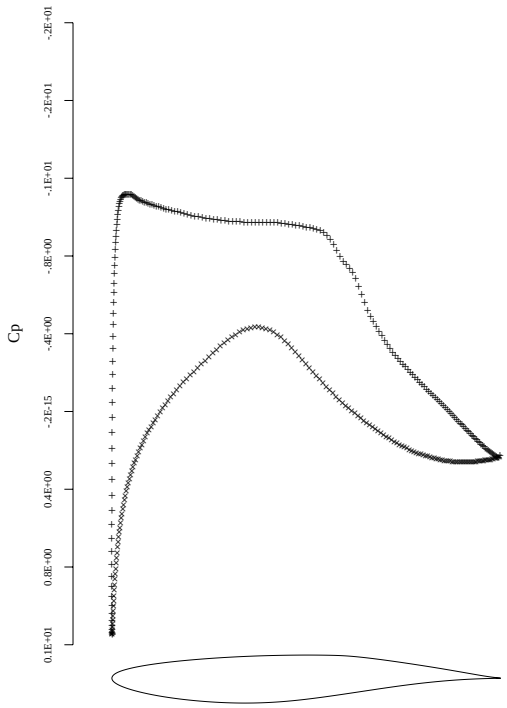


40a: Final Design Based on Total Drag Minimization

Figure 4.40: Drag Minimization of RAE Airfoil using the Continuous Adjoint Formulation. Grid - 512 x 64, $M = 0.75$, Fixed $C_l = 0.65$, $\alpha = 1$ degrees

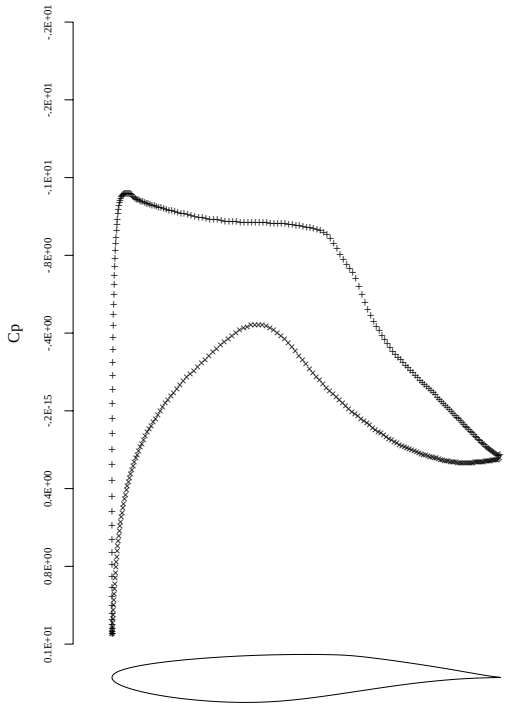


41a: Initial Solution of RAE Airfoil



41b: Final Design Based on Pressure Drag Minimization

Figure 4.41: Drag Minimization of RAE Airfoil using the Discrete Adjoint Formulation. Grid - 512 x 64, $M = 0.75$, Fixed $C_l = 0.65$, $\alpha = 1$ degrees



42a: Final Design Based on Total Drag Minimization

Figure 4.42: Drag Minimization of RAE Airfoil using the Discrete Adjoint Formulation. Grid - 512 x 64, $M = 0.75$, Fixed $C_l = 0.65$, $\alpha = 1$ degrees

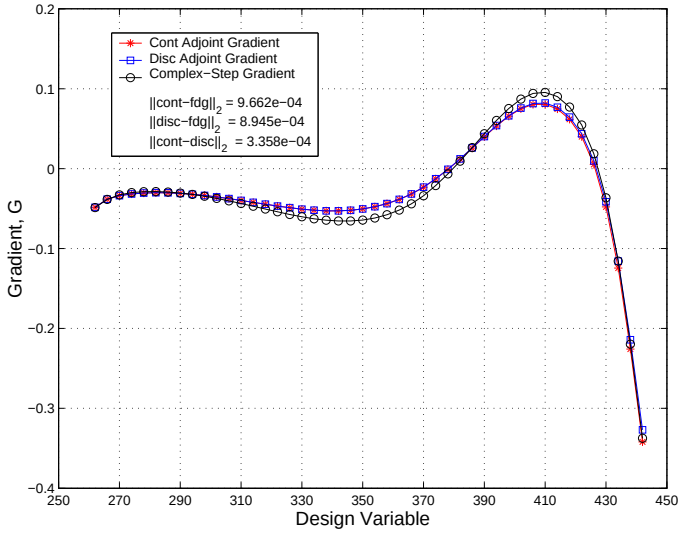


Figure 4.43: Adjoint Versus Complex-Step Gradients for Pressure Drag Minimization at Fixed C_l. Fine Grid - 512 x 64, $M = 0.75$, $C_l = 0.65$

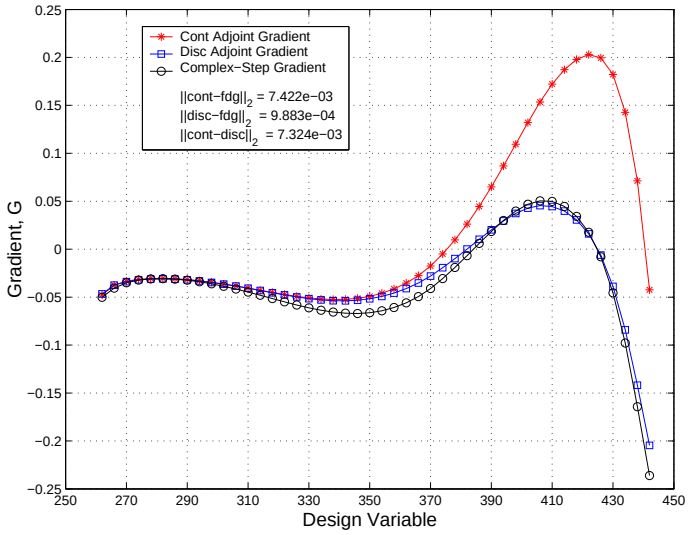


Figure 4.44: Adjoint Versus Complex-Step Gradients for Total Drag Minimization at Fixed C_l. Fine Grid - 512 x 64, $M = 0.75$, $C_l = 0.65$

Chapter 5

Optimum Shape Design for Unsteady Flows

This chapter presents an adjoint method for optimum shape design for unsteady flows. The goal is to develop a set of time accurate continuous and discrete adjoint equations and their corresponding boundary conditions. First, the complete general formulation of the time dependent optimal design problem is presented. Second, the time accurate continuous and discrete adjoint equations are derived. Third, results that demonstrate the application of the theory to two-dimensional oscillating airfoils are presented. Although the design procedure generates a single optimum airfoil shape, a similar method could be used to determine optimal time varying shapes if it were feasible to manufacture continuously deformable structures.

5.1 Governing Equations

As in the case of steady flow, it is convenient to use a body fitted coordinate system so that the computational domain is fixed. This requires the formulation of the Euler equations in a transformed coordinate system. The Cartesian coordinates and velocity components are denoted by x, y, and u, v. For a control volume Ω with a moving boundary $\partial\Omega$ moving with Cartesian velocity components x_t and y_t, the equations of

motion of the fluid can be written in integral form as

$$\frac{d}{dt}\iint_{\Omega}\mathbf{w}dxdy+\oint_{\partial\Omega}(\mathbf{f}dy-\mathbf{g}dx)=0, \tag{5.1}$$

where the state vector $\mathbf{w}$, inviscid flux vector $\mathbf{f}$ and $\mathbf{g}$ are described respectively by

$$w=\left\{\begin{array}{c}\rho\\ \rho u\\ \rho v\\ \rho E\end{array}\right\},\quad f=\left\{\begin{array}{c}\rho u\\ \rho u^2+p\\ \rho uv\\ \rho Eu+pu\end{array}\right\},\quad g=\left\{\begin{array}{c}\rho v\\ \rho uv\\ \rho v^2+p\\ \rho Ev+pv\end{array}\right\}. \tag{5.2}$$

In these definitions, ρ is the density and E is the total energy. The pressure is determined by the ideal gas equation of state

$$p=(\gamma-1)\,\rho\left[E-\frac{1}{2}\left(u_iu_i\right)\right].$$

For discussion of real applications using a discretization on a body conforming structured mesh, it is also useful to consider a transformation from the physical coordinates (x,y,t) to the computational coordinates (ξ_1,ξ_2,τ). Apply the following general transformation given by

$$\begin{aligned}\xi_i &= \xi_i\left(x_1,x_2,t\right), \quad \text{for } i=1,2\\ \tau &= t.\end{aligned}$$

The metrics can then be defined as

$$K_{ij}=\left[\frac{\partial x_i}{\partial\xi_j}\right],\quad J=\det\left(K\right),\quad K_{ij}^{-1}=\left[\frac{\partial\xi_i}{\partial x_j}\right].$$

Using the chain rule, the Euler equations can then be written in computational space as

$$\frac{\partial\left(Jw\right)}{\partial\tau}+\frac{\partial}{\partial\xi_i}\left[J\frac{\partial\xi_i}{\partial t}w+J\frac{\partial\xi_i}{\partial x_j}f_j\right]=0\quad\text{in }\mathcal{D}.$$

The mesh velocity in the computational domain, $\frac{\partial \xi_i}{\partial t}$, can be written in the form of the physical domain, by the following substitution

$$\frac{\partial (Jw)}{\partial \tau} + \frac{\partial}{\partial \xi_i}\left[J\frac{\partial \xi_i}{\partial x_j}\frac{\partial x_j}{\partial t}w + J\frac{\partial \xi_i}{\partial x_j}f_j \right] = 0.$$

A further rearrangement of the bracketed terms produce the unsteady Euler equations

$$\frac{\partial (Jw)}{\partial \tau} + \frac{\partial F_i}{\partial \xi_i} = 0, \tag{5.3}$$

where the inviscid flux contributions are now defined with respect to the computational cell faces by

$$F_i = S_{ij}\left[f_j - x_{t_j} w \right].$$

The quantity $S_{ij} = JK_{ij}^{-1}$ represents the projection of the ξ_i cell face along the x_j axis and the quantity x_{t_j} represents the mesh velocity in the jth coordinate direction.

5.2 Numerical Discretization

In this section, the numerical discretization of equation (5.3) will be discussed. Since the governing equation now models the unsteady flow field, it must be solved in a time accurate approach. This requires both the inclusion of the time varying terms in the fluxes and a proper discretization of the time derivative.

We will first re-examine at the spatial discretization of the total flux across the cell face. The Euler flux vector from equation (5.2) must be redefined to include the mesh velocity terms. The new Euler convective flux vectors can be written as

$$f = \begin{Bmatrix} \rho(u - x_t) \\ \rho u(u - x_t) + p \\ \rho v(u - x_t) \\ \rho E(u - x_t) + pu \end{Bmatrix} \text{ and } g = \begin{Bmatrix} \rho(v - x_t) \\ \rho u(v - x_t) \\ \rho v(v - x_t) + p \\ \rho E(v - x_t) + pv \end{Bmatrix}. \tag{5.4}$$

The above modifications can be incorporated into an existing steady Euler code with

very few modifications. Let us define the new flux velocity as

$$\widehat{q^-} = q^- - q_{mesh} \text{ and } \widehat{q^+} = q^+ - q_{mesh},$$

where the flux velocities, q^- and q^+, are the original Euler flux velocities and follow that of section 2.2.1. The mesh flux velocities are defined in a similar fashion as follows

$$q_{mesh} = y_{\eta_{i+\frac{1}{2},j}} x_{t_{ij}} - x_{\eta_{i+\frac{1}{2},j}} y_{t_{ij}},$$

where $x_{t_{ij}}$ and $y_{t_{ij}}$ are the mesh velocities in the x and y directions. When equation (5.3) is formulated for each computational cell a system of first order ordinary differential equations is obtained. To eliminate odd-even decoupling of the solution and overshoots before and after shock waves, the conservative fluxes are augmented by the artificial dissipation terms, as before.

5.2.1 Discretization of the Time Derivative Term

We now discuss the discretization of the time derivative term. For the solution of the steady-state Euler equations, the equation is marched until the time derivative vanishes. However, in an unsteady flow environment the equation must be marched in a time accurate approach. This can be accomplished either with explicit or implicit schemes. Implicit schemes generally require the inversion of a large matrix. The computational cost of an inversion is proportional to the size of the mesh and bandwidth of the matrix. Iterative methods have lower cost per iteration but generally require more iterations to produce a converged solution. For a detailed overview of the difference between each approach, please refer to Alonso [1]. In this research, a combination of both an explicit and implicit approach first proposed by Jameson [36] is employed. The time dependent flow equation is discretized with an implicit scheme, but the resulting equations are solved at each time step using an iterative algorithm that is generally used for explicit schemes. To obtain a fully implicit algorithm we approximate (5.3) as

$$\frac{d}{d\tau}\left[w_{i,j}^{n+1} V_{i,j}^{n+1}\right] + R(w_{i,j}^{n+1}) = 0. \tag{5.5}$$

The time derivative term can be approximated by a k th-order implicit backward difference formula (BDF) such as,

$$\frac{d}{d\tau} = \frac{1}{\Delta\tau} \sum_{q=1}^{k} \frac{1}{q} \left[\Delta^-\right]^q, \tag{5.6}$$

where

$$\Delta^- = w_{i,j}^{n+1} - w_{i,j}^{n}.$$

Following Melson et al. [58], the physical time derivative can be expressed as

$$\begin{aligned} \frac{d}{d\tau}\left[w_{i,j}^{n+1} V_{i,j}^{n+1}\right] &= \frac{1}{\Delta\tau}\left[\sum_{m=0}^{M} a_m w^{(n+1-m)} V^{(n+1-m)}\right] \\ &= \frac{1}{\Delta\tau}\left[a_0 w^{(n+1)} V^{(n+1)}\right. \\ &\quad \left. + E\left(w^{(n)}V^{(n)}, w^{(n-1)}V^{(n-1)}, \cdots, w^{(n+1-M)}V^{(n+1-M)}\right)\right], \end{aligned}$$

where the operator E represents the terms of the time derivative for previous time steps. A second order accurate backward difference expansion of the above equation produces an A-stable scheme. The stability region of an A-stable scheme covers the entire left half of the complex plane. Consequently the time step can be chosen arbitrarily large, subject to the requirement of sufficient accuracy. Dahlquist et al. [14] proved that A-stable multistep schemes are no better than second order accurate. A third order expansion of the backward difference formula generates a stiffly stable scheme as reported by Gear [22] and Lambert [48]. In this case the stability region covers most but not all of the left half of the complex plane. A subset of the high frequency error modes may not be damped, but the inclusion of artificial dissipation fluxes eliminates these error components. Alonso [1] has shown that this scheme is robust in practice. The fourth order expansion also generates a stiffly stable scheme, but with more undamped high frequency error modes. In practice this scheme has proved to be robust for only a selected number of problems.

In this work, a second order expansion of equation (5.6) is employed as shown below

$$\frac{3}{2\Delta\tau}\left[w_{i,j}^{n+1}V_{i,j}^{n+1}\right] - \frac{2}{\Delta\tau}\left[w_{i,j}^{n}V_{i,j}^{n}\right] + \frac{1}{2\Delta\tau}\left[w_{i,j}^{n-1}V_{i,j}^{n-1}\right] + R(w_{i,j}^{n+1}) = 0. \tag{5.7}$$

Equation (5.7) represents an implicit set of coupled ordinary differential equations and can be solved at each time step using the explicit multistage modified Runge-Kutta scheme. We define a new modified residual $R^*(w_{i,j})$ as

$$R^*(w_{i,j}) = \frac{3}{2\Delta\tau}\left[w_{i,j}^{n+1}V_{i,j}^{n+1}\right] - \frac{2}{\Delta\tau}\left[w_{i,j}^{n}V_{i,j}^{n}\right] + \frac{1}{2\Delta\tau}\left[w_{i,j}^{n-1}V_{i,j}^{n-1}\right] + R(w_{i,j}^{n+1}). \tag{5.8}$$

The modified residual is then marched to steady-state in a fictitious time, t^*, as follows

$$\frac{dw_{i,j}}{dt^*} + R^*(w_{i,j}) = 0.$$

5.3 Reduced Frequency

The unsteady optimization method in this work is applied to the reduction of the time-averaged drag of an airfoil pitching about its quarter chord. An important dimensionless parameter that describes the level of unsteadiness is the reduced frequency, defined as

$$\omega_r = \frac{\theta c}{2V},$$

where θ is the angular frequency of the oscillation, c is the airfoil chord, and V the flow field freestream velocity. For $\omega_r = 0$, the flow is steady. For $0 \leq \omega_r \leq 0.05$ the flow is defined as quasi-steady. In a quasi-steady flow environment, the unsteady effects are negligible. An example of a quasi-steady flow is an airfoil oscillating at a very slow speed, such that the instantaneous flow solution is similar to the steady-state solution. Flows with $\omega_r \geq 0.05$ are defined as unsteady, while flows with $\omega_r \geq 0.2$ is termed highly unsteady.

5.4 General Formulation

As in Chapter 4, the aerodynamic properties that define the cost function are functions of the flow field variables, w, and the physical location of the boundary, which may be represented by the function f. We then introduce the cost function

$$I = \int_0^{t_f} \mathcal{L}(w, f)dt + \mathcal{M}(w(t_f)), \tag{5.9}$$

where the function $\mathcal{L}$ depends on the flow solution w, and the shape function f and the function $\mathcal{M}$ depends on the time dependent flow solution.

A change in f results in a change

$$\delta I = \int_0^{t_f} \left(\frac{\partial \mathcal{L}^T}{\partial w}\delta w + \frac{\partial \mathcal{L}^T}{\partial f}\delta f \right) dt + \frac{\partial \mathcal{M}^T}{\partial w}\delta w(t_f) \tag{5.10}$$

in the cost function. Using control theory the governing equations of the flow field are now introduced as a constraint in such a way that the final expression for the gradient does not require reevaluation of the flow field. In order to achieve this, δw must be eliminated from equation (5.10). Suppose that the following equation defines the time dependent flow solution

$$\frac{\partial w}{\partial t} + R(w, f) = 0,$$

where R represents a residue containing the convective and dissipative fluxes. The variation in the flow solution is

$$\frac{\partial}{\partial t}\delta w + \left[\frac{\partial R}{\partial w}\right] \delta w + \left[\frac{\partial R}{\partial f}\right] \delta f = 0.$$

Next, introduce a Lagrange multiplier ψ to the time dependent flow equation and integrate it over time to yield.

$$\int_0^{t_f} \psi^T \left(\frac{\partial}{\partial t}\delta w + \left[\frac{\partial R}{\partial w}\right] \delta w + \left[\frac{\partial R}{\partial f}\right] \delta f \right) = 0. \tag{5.11}$$

Subtract equation (5.11) from the variation of the cost function to arrive at the following equation:

$$\begin{aligned}\delta I = & \int_0^{t_f} \left(\frac{\partial \mathcal{L}^T}{\partial w} \delta w + \frac{\partial \mathcal{L}^T}{\partial f} \delta f \right) dt + \frac{\partial \mathcal{M}^T}{\partial w} \delta w(t_f) \\ - & \int_0^{t_f} \psi^T \left(\frac{\partial}{\partial t} \delta w + \left[\frac{\partial R}{\partial w} \right] \delta w + \left[\frac{\partial R}{\partial f} \right] \delta f \right) dt. \end{aligned} \tag{5.12}$$

Next separate the $\int_0^{t_f} \psi^T \frac{\partial}{\partial t} \delta w dt$ term from the integral over time of the product of the Lagrange multiplier ψ and the time dependent flow equation. Then collect the δw and δf terms and rearrange them in equation (5.12) to produce the following expression:

$$\begin{aligned}\delta I = & \int_0^{t_f} \left(\frac{\partial \mathcal{L}^T}{\partial w} - \psi^T \left[\frac{\partial R}{\partial w} \right] \right) \delta w dt + \int_0^{t_f} \left(\frac{\partial \mathcal{L}^T}{\partial f} - \psi^T \left[\frac{\partial R}{\partial f} \right] \right) \delta f dt \\ - & \int_0^{t_f} \psi^T \frac{\partial}{\partial t} \delta w dt + \frac{\partial \mathcal{M}^T}{\partial w} \delta w(t_f). \end{aligned}$$

Lastly, integrate the term $\int_0^{t_f} \psi^T \frac{\partial}{\partial t} \delta w dt$ by parts, to yield

$$\begin{aligned}\delta I = & \int_0^{t_f} \left(\frac{\partial \mathcal{L}^T}{\partial w} + \frac{\partial \psi^T}{\partial t} - \psi^T \left[\frac{\partial R}{\partial w} \right] \right) \delta w dt + \left(\frac{\partial \mathcal{M}^T}{\partial w} - \psi^T(t_f) \right) \delta w(t_f) \\ + & \int_0^{t_f} \left(\frac{\partial \mathcal{L}^T}{\partial f} - \psi^T \left[\frac{\partial R}{\partial f} \right] \right) \delta f dt. \end{aligned}$$

Choose ψ to satisfy the adjoint equation

$$\frac{\partial \psi}{\partial t} - \left[\frac{\partial R}{\partial w} \right]^T \psi = - \frac{\partial \mathcal{L}}{\partial w}$$

with the terminal boundary condition

$$\psi(t_f) = \frac{\partial \mathcal{M}}{\partial w}.$$

Then the variation of the cost function reduces to

$$\delta I = \mathcal{G}^T \delta f,$$

where

$$\mathcal{G}^{\mathcal{T}} = \int_0^{t_f} \left(\frac{\partial \mathcal{L}^T}{\partial f} - \psi^T \frac{\partial R}{\partial f} \right) dt.$$

Optimal control of time dependent trajectories is generally complicated by the need to solve the adjoint equation in reverse time from a final boundary condition using data from the trajectory solution, which in turn depends on the control derived from the adjoint solution. The sensitivities are determined by the solution of the adjoint equation in reverse time from the terminal boundary condition and the time dependent solution of the flow equation. These sensitivities are then used to get a direction of improvement and steps are taken until convergence is achieved. Note here that the difference between the gradient for a multipoint design approach and the time dependent problem is minimal, where the integral over time in the time dependent gradient can be replaced by a summation over the various design cases for a multipoint design approach. The equations are identical if a constant time step is used for the time dependent approach. However, the flow solution used for the calculation of the gradient for the time dependent problem would be based on the solution of the unsteady problem instead of the steady solution at each time instance for the multipoint approach. The computational costs of unsteady optimization problems are directly proportional to the desired number of time steps. The unsteady flow calculation can be obtained either by the use of implicit time stepping schemes or a nonlinear frequency domain approach [57].

5.5 Time Accurate Continuous Adjoint Equations

To control the surface pressure by varying the airfoil shape, it is convenient to retain a fixed computational domain. Variations in the shape then result in corresponding variations in the mapping derivatives defined by K. The cost function for drag minimization is

$$\begin{aligned} I = \int_0^{t_f} C_d \, dt &= \int_0^{t_f} C_a \cos\alpha + C_n \sin\alpha \, dt \\ &= \frac{1}{\frac{1}{2}\gamma P_\infty M_\infty^2 \bar{c}} \int_0^{t_f}\int_{\mathcal{B}} p \left(\frac{\partial y}{\partial \xi}\cos\alpha - \frac{\partial x}{\partial \xi}\sin\alpha \right) d\xi dt, \end{aligned}$$

where I is the total drag of the time accurate problem, and C_a and C_n are the axial and normal force coefficients respectively. The design problem is now treated as a control problem where the control function is the airfoil shape, which is chosen to minimize I subject to the constraints defined by the flow equations. A variation in the shape causes a variation δp in the pressure and consequently a variation in the cost function

$$\begin{aligned} \delta I &= \frac{1}{\frac{1}{2}\gamma P_\infty M_\infty^2 \bar{c}} \int_0^{t_f}\int_{\mathcal{B}} \delta p \left[\frac{\partial y}{\partial \xi}\cos\alpha - \frac{\partial x}{\partial \xi}\sin\alpha \right] \\ &\qquad + p \left[\delta\left(\frac{\partial y}{\partial \xi} \right)\cos\alpha - \delta\left(\frac{\partial x}{\partial \xi} \right)\sin\alpha \right] d\xi dt. \end{aligned} \tag{5.13}$$

Since the pressure, p, depends on the state vector, w, through the equation of state, then the variation δp is determined from the variation δw. Define the time dependent flow equation as

$$\frac{\partial w}{\partial t} + \frac{\partial F_k}{\partial \xi_k} = 0.$$

Define the Euler Jacobian matrices as

$$A_k = \frac{\partial f_k}{\partial w}, \quad C_k = S_{kl} A_l.$$

Then the variation in the flow solution can be written as

$$\frac{\partial}{\partial t}\delta w + \frac{\partial}{\partial \xi_k}\delta F_k = 0,$$

where

$$\delta F_k = C_k \delta w + \delta S_{kl} f_l.$$

Multiplying by a co-state vector ψ, also known as a Lagrange Multiplier, and integrating over the space and time produces

$$\int_0^{t_f}\int_{\mathcal{D}} \psi^T \left[\frac{\partial}{\partial t}\delta w + \frac{\partial}{\partial \xi_k}\left(C_k \delta w + \delta S_{kl} f_l\right)\right] d\mathcal{D}dt = 0.$$

Separate the equation into two terms and switch the order of the domain and time integrals for the first term to yield

$$\int_{\mathcal{D}}\int_0^{t_f} \psi^T \frac{\partial}{\partial t}\delta w dt d\mathcal{D} + \int_0^{t_f}\int_{\mathcal{D}} \psi^T \frac{\partial}{\partial \xi_k}\left(C_k \delta w + \delta S_{kl} f_l\right) d\mathcal{D}dt = 0.$$

If ψ is differentiable, then the two terms in the above equation can be integrated by parts to give

$$\begin{aligned} &\int_{\mathcal{D}} \left(\left[\psi^T \delta w\right]_0^{t_f} - \int_0^{t_f} \frac{\partial \psi^T}{\partial t}\delta w dt \right) d\mathcal{D} \\ + &\int_0^{t_f} \left[\int_{\mathcal{B}} n_k \psi^T \left(C_k \delta w + \delta S_{kl} f_l\right) d\mathcal{B} - \int_{\mathcal{D}} \frac{\partial \psi^T}{\partial \xi_k}\left(C_k \delta w + \delta S_{kl} f_l\right) d\mathcal{D} \right] dt = 0. \end{aligned}$$

The next procedure is to rearrange the terms in the equation such that integrands that are multiplied by the variation of the state vector, δw, are grouped together and terms that are multiplied by the variation of the metric terms are separated into a different integral. This procedure is crucial to isolate the integral that will produce

the time accurate continuous adjoint equation.

$$\begin{aligned} & \int_{\mathcal{D}} \left[\psi^T(t_f)\delta w(t_f) - \psi^T(0)\delta w(0)\right] d\mathcal{D} \\ - & \int_0^{t_f}\int_{\mathcal{D}} \frac{\partial \psi^T}{\partial t}\delta w + \frac{\partial \psi^T}{\partial \xi_k} C_k \delta w d\mathcal{D}dt + \int_0^{t_f}\int_{\mathcal{B}} n_k \psi^T \delta F_k d\mathcal{B} \\ + & \int_0^{t_f}\left[\int_{\mathcal{B}} n_k \psi^T \delta S_{kl} f_l d\mathcal{B} - \int_{\mathcal{D}} \frac{\partial \psi^T}{\partial \xi_k}\delta S_{kl} f_l d\mathcal{D}\right] dt = 0. \end{aligned}$$

Since the left-hand expression equals zero, it may be subtracted from the variation of the cost function (5.13) to give

$$\begin{aligned} \delta I = & \frac{1}{\frac{1}{2}\gamma P_\infty M_\infty^2 \bar{c}} \int_0^{t_f}\int_{\mathcal{B}_W} \delta p \left[\frac{\partial y}{\partial \xi}\cos\alpha - \frac{\partial x}{\partial \xi}\sin\alpha\right] \\ & + p\left[\delta\left(\frac{\partial y}{\partial \xi}\right)\cos\alpha - \delta\left(\frac{\partial x}{\partial \xi}\right)\sin\alpha\right] d\xi dt \\ & - \int_{\mathcal{D}} \left[\psi^T(t_f)\delta w(t_f) - \psi^T(0)\delta w(0)\right] d\mathcal{D} + \int_0^{t_f}\int_{\mathcal{D}} \left[\frac{\partial \psi^T}{\partial t} + \frac{\partial \psi^T}{\partial \xi_k} C_k\right]\delta w d\mathcal{D}dt \\ & - \int_0^{t_f}\int_{\mathcal{B}} n_k \psi^T \delta F_k d\mathcal{B} - \int_0^{t_f}\left[\int_{\mathcal{B}} n_k \psi^T \delta S_{kl} f_l d\mathcal{B} - \int_{\mathcal{D}} \frac{\partial \psi^T}{\partial \xi_k}\delta S_{kl} f_l d\mathcal{D}\right] dt. \end{aligned} \quad (5.14)$$

Now, since ψ is an arbitrary differentiable function, it may be chosen in such a way that δI no longer depends explicitly on the variation of the state vector δw. The gradient of the cost function can then be evaluated directly from the metric variations without having to re-compute the variation δw resulting from the perturbation of each design variable. The variation δw can then be eliminated by solving for the Lagrange Multiplier, ψ, by setting the transpose of the integrand of the second integral in the third line of equation (5.14) to zero to produce a differential adjoint system governing ψ

$$\frac{\partial \psi}{\partial t} + C_k^T \frac{\partial \psi}{\partial \xi_k} = 0 \quad \text{in D.} \quad (5.15)$$

At the outer boundary incoming characteristics for ψ corresponds to outgoing characteristics for δw. Consequently we can choose boundary conditions for ψ such that

$$n_k \psi^T C_k \delta w = 0.$$

If the coordinate transformation is such that δS is negligible in the far-field, then the only remaining boundary term is

$$-\int_{\mathcal{B}_W} \psi^T \delta F_2 d\xi_1.$$

Thus, by letting ψ satisfy the boundary condition, then

$$\psi_j n_j = \frac{1}{\frac{1}{2}\gamma P_\infty M_\infty^2 \bar{c}} \left[\frac{\partial y}{\partial \xi} \cos\alpha - \frac{\partial x}{\partial \xi} \sin\alpha \right] \quad \text{on } \mathcal{B}_W, \tag{5.16}$$

where n_j are the components of the surface normal. Since the initial condition for the Lagrange multipliers are set to zero, then

$$\psi^T(0)\delta w(0) = 0.$$

Since the problem is periodic in nature and that the cost function used for this problem is not dependent upon t_f, then

$$\psi^T(t_f)\delta w(t_f) = 0.$$

Equation (5.14) finally reduces to the following

$$\begin{aligned} \delta I &= \frac{1}{\frac{1}{2}\gamma P_\infty M_\infty^2 \bar{c}} \int_0^{t_f}\int_{\mathcal{B}_W} p \left[\delta\left(\frac{\partial y}{\partial \xi}\right) \cos\alpha - \delta\left(\frac{\partial x}{\partial \xi}\right) \sin\alpha \right] d\xi dt \\ &\quad - \int_0^{t_f} \left[\int_{\mathcal{B}} n_k \psi^T \delta S_{kl} f_l d\mathcal{B} - \int_{\mathcal{D}} \frac{\partial \psi^T}{\partial \xi_k} \delta S_{kl} f_l d\mathcal{D} \right] dt. \end{aligned}$$

The above equation is then used to solve for the gradient, which can then provide a direction of improvement to reduce the objective function.

5.6 Time Accurate Discrete Adjoint Equations

As in the case of steady flow, the time accurate discrete adjoint equation is obtained by applying control theory directly to the set of time accurate discrete field equations.

The resulting equation depends on the type of scheme used to solve the flow equations. This work uses a cell-centered multigrid-driven fully implicit scheme with upwind-biased blended first and third order fluxes as the artificial dissipation scheme.

To develop the time accurate discrete adjoint equation, the first step is to take a variation of the modified residual represented in equation (5.8) with respect to the state vector, w and shape function, f (only terms that are multiplied by δw are shown) to produce

$$\begin{aligned} \delta R_{i,j}^{*^{n+1}}(w) &= \frac{3}{2\Delta t}\left[\delta w_{i,j}^{n+1} V_{i,j}^{n+1}\right] - \frac{2}{\Delta t}\left[\delta w_{i,j}^{n} V_{i,j}^{n}\right] \\ &\quad + \frac{1}{2\Delta t}\left[\delta w_{i,j}^{n-1} V_{i,j}^{n-1}\right] + \delta R_{i,j}^{n+1}(w). \end{aligned} \tag{5.17}$$

Multiply the above equation by the transpose of the Lagrange Multiplier and sum over the domain and time to yield

$$\begin{aligned} \sum_{t=0}^{t_f}\sum_{\Omega} \psi_{i,j}^{T} \delta R_{i,j}^{*}(w) = \cdots &+ \psi_{i,j}^{T^{n+1}} \delta R_{i,j}^{*^{n+1}}(w) \\ &+ \psi_{i,j}^{T^{n+2}} \delta R_{i,j}^{*^{n+2}}(w) + \psi_{i,j}^{T^{n+3}} \delta R_{i,j}^{*^{n+3}}(w) + \cdots \end{aligned}$$

Substitute equation (5.17) into the $(n+1)$, $(n+2)$, and $(n+3)$ terms of the modified residual in the above equation to yield

$$\begin{aligned} \sum_{t=0}^{t_f}\sum_{\Omega} \psi_{i,j}^{T} \delta R_{i,j}^{*}(w) = \cdots &+ \psi_{i,j}^{T^{n+1}} \left[\frac{3}{2\Delta t} V^{n+1} \delta w_{i,j}^{n+1} - \frac{2}{\Delta t} V^{n} \delta w_{i,j}^{n}(w) \right. \\ &\qquad \left. + \frac{1}{2\Delta t} V^{n-1} \delta w_{i,j}^{n-1} + \delta R_{i,j}^{n+1}\right] \\ &+ \psi_{i,j}^{T^{n+2}} \left[\frac{3}{2\Delta t} V^{n+2} \delta w_{i,j}^{n+2} - \frac{2}{\Delta t} V^{n+1} \delta w_{i,j}^{n+1}(w) \right. \\ &\qquad \left. + \frac{1}{2\Delta t} V^{n} \delta w_{i,j}^{n} + \delta R_{i,j}^{n+2}\right] \\ &+ \psi_{i,j}^{T^{n+3}} \left[\frac{3}{2\Delta t} V^{n+3} \delta w_{i,j}^{n+3} - \frac{2}{\Delta t} V^{n+2} \delta w_{i,j}^{n+2}(w) \right. \\ &\qquad \left. + \frac{1}{2\Delta t} V^{n+1} \delta w_{i,j}^{n+1} + \delta R_{i,j}^{n+3}\right] + \cdots . \end{aligned}$$

Keeping only the $(n+1)$ terms, rearrange the terms in the equation to produce the time accurate discrete adjoint equation

$$\sum_{t=0}^{t_f}\sum_{\Omega}\psi_{i,j}^{T}\delta R_{i,j}^{*}(w)=\cdots \quad + \quad \left[\frac{3}{2\Delta t}\psi_{i,j}^{T^{n+1}}-\frac{2}{\Delta t}\psi_{i,j}^{T^{n+2}}+\frac{1}{2\Delta t}\psi_{i,j}^{T^{n+3}}\right]V^{n+1}\delta w_{i,j}^{n+1} + \psi_{i,j}^{T^{n+1}}\delta R_{i,j}^{n+1}+\cdots . \tag{5.18}$$

Next we introduce the discrete cost function for the drag minimization problem as

$$\begin{aligned} I_c=\sum_{t=0}^{t_f}C_d\,\Delta t &= \sum_{t=0}^{t_f}\left(C_a\cos\alpha+C_n\sin\alpha\right)\Delta t \\ &= \frac{1}{\frac{1}{2}\gamma P_\infty M_\infty^2\bar{c}}\sum_{t=0}^{t_f}\sum_{i=LTE}^{UTE}p_{i,W}\left(\frac{\Delta y_i}{\Delta s_i}\cos\alpha-\frac{\Delta x_i}{\Delta s_i}\sin\alpha\right)\Delta s_i\Delta t, \end{aligned}$$

where LTE is the lower trailing edge, UTE is the upper trailing edge, Δs_i is the surface cell arc length, and $\Delta p_{i,W}$ is the wall pressure. In this research the wall pressure is defined as such

$$p_{i,W}=\frac{1}{2}\left(p_{i,2}+p_{i,1}\right),$$

where $p_{i,2}$ is the value of the pressure in the cell above the wall and $p_{i,1}$ is the value of the pressure in the cell below the wall. A variation in the cost function will result in a variation, Δp, in the pressure and variations, Δy and Δx, in the geometry. The variation of the cost function for drag minimization can be written as

$$\begin{aligned} \delta I_c &= \frac{1}{\frac{1}{2}\gamma P_\infty M_\infty^2\bar{c}}\sum_{t=0}^{t_f}\left[\sum_{i=LTE}^{UTE}\frac{1}{2}\left(\frac{\Delta y_i}{\Delta s_i}\cos\alpha-\frac{\Delta x_i}{\Delta s_i}\sin\alpha\right)\frac{\partial p}{\partial w}\left(\delta w_{i,2}+\delta w_{i,1}\right)\Delta s_i\right. \\ &\quad \left.+\sum_{i=LTE}^{UTE}\left(\frac{1}{2}\left(p_{i,2}+p_{i,1}\right)-p_\infty\right)\left[\cos\alpha\;\delta\left(\Delta y_i\right)-\sin\alpha\;\delta\left(\Delta x_i\right)\right]\right]\Delta t. \end{aligned} \tag{5.19}$$

The time dependent discrete Euler equations can now be introduced into δI as a constraint to produce

$$\delta I=\delta I_c-\sum_{t=0}^{t_f}\sum_{\Omega}\psi_{i,j}^{T}\delta R_{i,j}^{*}(w).$$

Substitute equation (5.18) and (5.19) into the above expression to yield

$$\begin{aligned}\delta I &= \frac{1}{\frac{1}{2}\gamma P_\infty M_\infty^2 \bar{c}} \sum_{t=0}^{t_f} \left[\sum_{i=LTE}^{UTE} \frac{1}{2} \left(\frac{\Delta y_i}{\Delta s_i} \cos\alpha - \frac{\Delta x_i}{\Delta s_i} \sin\alpha \right) \frac{\partial p}{\partial w} \left(\delta w_{i,2} + \delta w_{i,1} \right) \Delta s_i \right. \\ &\qquad \left. + \sum_{i=LTE}^{UTE} \left(\frac{1}{2} \left(p_{i,2} + p_{i,1} \right) - p_\infty \right) \left[\cos\alpha \; \delta \left(\Delta y_i \right) - \sin\alpha \; \delta \left(\Delta x_i \right) \right] \right] \Delta t \\ &\quad - \left[\cdots + \left[\frac{3}{2\Delta t} \psi_{i,j}^{T^{n+1}} - \frac{2}{\Delta t} \psi_{i,j}^{T^{n+2}} + \frac{1}{2\Delta t} \psi_{i,j}^{T^{n+3}} \right] V^{n+1} \delta w_{i,j}^{n+1} \right. \\ &\qquad \left. + \psi_{i,j}^{T^{n+1}} \delta R_{i,j}^{n+1} + \cdots \right].\end{aligned}$$

The above expression can be rearranged into two main categories: first, terms that are multiplied by the variation of the state vector, δw; and second, terms that are multiplied by the variation of the shape function, δf. The rearranged equation can be expressed as

$$\begin{aligned}\delta I = \cdots \; &+ \; \frac{1}{2} \left(\frac{\Delta y_i}{\Delta s_i} \omega \cos\alpha - \frac{\Delta x_i}{\Delta s_i} \omega \sin\alpha \right) \frac{\partial p}{\partial w} \left(\delta w_{i,2}^{n+1} + \delta w_{i,1}^{n+1} \right) \Delta s_i \Delta t \\ &- \; \left[\frac{3}{2\Delta t} \psi_{i,j}^{T^{n+1}} - \frac{2}{\Delta t} \psi_{i,j}^{T^{n+2}} + \frac{1}{2\Delta t} \psi_{i,j}^{T^{n+3}} \right] V^{n+1} \delta w_{i,j}^{n+1} - \psi_{i,j}^{T^{n+1}} \delta_w R_{i,j}^{n+1} \\ &+ \; \left(\frac{1}{2} \left(p_{i,2} + p_{i,1} \right) - p_\infty \right) \left[\delta \left(\Delta y_i \right) \omega \cos\alpha - \delta \left(\Delta x_i \right) \omega \sin\alpha \right] \Delta t \\ &- \; \psi_{i,j}^{T^{n+1}} \delta_f R_{i,j}^{n+1} + \cdots, \end{aligned} \tag{5.20}$$

where $\omega = \frac{1}{\frac{1}{2}\gamma P_\infty M_\infty^2 \bar{c}}$, $\delta_w R_{i,j}^{n+1}$ are terms that are a result of taking a variation of the residual with respect to the state vector, w, and $\delta_f R_{i,j}^{n+1}$ are terms that are a result of taking a variation of the residual with respect to the shape function, f. Equation (5.20) can be further expanded to isolate the $\delta w_{i,2}^{n+1}$ terms. This step is needed to produce the boundary condition source term for the time accurate discrete adjoint equation. The extraction of the $\delta w_{i,2}^{n+1}$ term results in the following split in

the equation

$$\begin{aligned}
\delta I = \cdots \quad &+ \quad \frac{1}{2}\left(\frac{\Delta y_i}{\Delta s_i}\omega\cos\alpha - \frac{\Delta x_i}{\Delta s_i}\omega\sin\alpha\right)\frac{\partial p}{\partial w}\delta w_{i,1}^{n+1}\Delta s_i\Delta t \\
&+ \quad \frac{1}{2}\left(\frac{\Delta y_i}{\Delta s_i}\omega\cos\alpha - \frac{\Delta x_i}{\Delta s_i}\omega\sin\alpha\right)\frac{\partial p}{\partial w}\delta w_{i,2}^{n+1}\Delta s_i\Delta t \\
&- \quad \left[\frac{3}{2\Delta t}\psi_{i,2}^{T^{n+1}} - \frac{2}{\Delta t}\psi_{i,2}^{T^{n+2}} + \frac{1}{2\Delta t}\psi_{i,2}^{T^{n+3}}\right]V^{n+1}\delta w_{i,2}^{n+1} - \psi_{i,2}^{T^{n+1}}\delta_w R_{i,2}^{n+1} \\
&- \quad \left[\frac{3}{2\Delta t}\psi_{i,j}^{T^{n+1}} - \frac{2}{\Delta t}\psi_{i,j}^{T^{n+2}} + \frac{1}{2\Delta t}\psi_{i,j}^{T^{n+3}}\right]V^{n+1}\delta w_{i,j}^{n+1} - \psi_{i,j}^{T^{n+1}}\delta_w R_{i,j}^{n+1} \\
&+ \quad \left(\frac{1}{2}\left(p_{i,2}+p_{i,1}\right) - p_\infty\right)\left[\delta\left(\Delta y_i\right)\omega\cos\alpha - \delta\left(\Delta x_i\right)\omega\sin\alpha\right]\Delta t \\
&- \quad \psi_{i,j}^{T^{n+1}}\delta_f R_{i,j}^{n+1} + \cdots
\end{aligned} \tag{5.21}$$

A brief explanation of the steps taken in the above equation is needed here. First, the first line in equation (5.20) is split into $\delta w_{i,1}^{n+1}$ and $\delta w_{i,2}^{n+1}$ resulting in the first and second line in the above equation. Second, $\delta w_{i,2}^{n+1}$ is extracted from the second lines in equation (5.20) to produce the third and fourth lines in the above equation. The last line in the above equation remains as is. To establish the drag minimization boundary condition source term for the time accurate discrete adjoint equation, we first need to expand the $\delta_w R_{i,2}^{n+1}$ term. From equation (4.26), the $\delta_w R_{i,2}^{n+1}$ term can be written as follows

$$\begin{aligned}
\psi_{i,2}^{T^{n+1}}\delta_w R_{i,2}^{n+1} = \quad - \quad \frac{1}{2}\Big[&A_{i-\frac{1}{2},2}^{T^{n+1}}\left(\psi_{i,2}^{n+1} - \psi_{i-1,2}^{n+1}\right) + A_{i+\frac{1}{2},2}^{T^{n+1}}\left(\psi_{i+1,2}^{n+1} - \psi_{i,2}^{n+1}\right) \\
&+B_{i,\frac{5}{2}}^{T^{n+1}}\left(\psi_{i,3}^{n+1} - \psi_{i,2}^{n+1}\right) \\
&-\Delta y_\xi \psi_{2_{i,2}}^{n+1} + \Delta x_\xi \psi_{3_{i,2}}^{n+1}\Big]\delta w_{i,2}^{n+1}.
\end{aligned}$$

The time accurate discrete adjoint equation can now be defined as

$$\frac{\partial \psi_{i,j}^{n+1}}{\partial \tau} - \left[\frac{3}{2\Delta t}\psi_{i,j}^{T^{n+1}} - \frac{2}{\Delta t}\psi_{i,j}^{T^{n+2}} + \frac{1}{2\Delta t}\psi_{i,j}^{T^{n+3}}\right]V^{n+1} - \psi_{i,j}^{T^{n+1}}\delta_w R_{i,j}^{n+1} = 0.$$

At cell $(i,2)$ the time accurate discrete adjoint equation is as follows,

$$\frac{\partial \psi_{i,2}^{n+1}}{\partial \tau} - \left[\frac{3}{2\Delta t}\psi_{i,j}^{T^{n+1}} - \frac{2}{\Delta t}\psi_{i,j}^{T^{n+2}} + \frac{1}{2\Delta t}\psi_{i,j}^{T^{n+3}}\right] V^{n+1} - \frac{1}{2}\left[A_{i-\frac{1}{2},2}^{T^{n+1}}\left(\psi_{i,2}^{n+1} - \psi_{i-1,2}^{n+1}\right)\right.$$
$$\left. + A_{i+\frac{1}{2},2}^{T^{n+1}}\left(\psi_{i+1,2}^{n+1} - \psi_{i,2}^{n+1}\right) + B_{i,\frac{5}{2}}^{T^{n+1}}\left(\psi_{i,3}^{n+1} - \psi_{i,2}^{n+1}\right) - \Phi\right], \tag{5.22}$$

where Φ is the source term for drag minimization,

$$\Phi = \Delta y_\xi \psi_{2_{i,2}}^{n+1} - \Delta x_\xi \psi_{3_{i,2}}^{n+1} + \left(\frac{\Delta y_i}{\Delta s_i}\omega\cos\alpha - \frac{\Delta x_i}{\Delta s_i}\omega\sin\alpha\right)\frac{\partial p}{\partial w}\Delta s_i \Delta t.$$

All the terms in equation (5.22) except for the source term scale as the square of Δx. Therefore, as the mesh width is reduced, the terms in the source term if divided by Δs_i must approach zero as the solution reaches a steady-state. One then recovers the continuous adjoint boundary condition as stated in equation (5.16). Thus equation (5.21) simplifies to the following equation,

$$\delta I = \cdots + \left(\frac{1}{2}\left(p_{i,2} + p_{i,1}\right) - p_\infty\right)\left[\delta\left(\Delta y_i\right)\omega\cos\alpha - \delta\left(\Delta x_i\right)\omega\sin\alpha\right]\Delta t - \psi_{i,j}^{T^{n+1}}\delta_f R_{i,j}^{n+1} + \cdots$$

The above equation represents the total gradient obtained using the time accurate discrete adjoint approach to reduce the total drag of a pitching airfoil.

5.7 Design Process

To understand the difference between optimizing an airfoil shape using either the full unsteady or steady approaches, four different design processes have been evaluated. These are the full unsteady design (unsteady-flow unsteady-adjoint), partial unsteady design (unsteady-flow steady-adjoint), time-averaged-flow steady-adjoint design, and multipoint design.

The four design processes have been applied to the redesign of the RAE 2822 and VR-7 airfoils. The cost function is the time-averaged drag coefficient. The design variables are the surface mesh points. The constraints are constant thickness to chord ratio and fixed time-averaged lift coefficient. The airfoil undergoes a forced

pitching oscillation about the quarter-chord. The angle of incidence is given by

$$\alpha(t) = \alpha_o + \alpha_m \sin(\omega t),$$

where $\alpha_o = 0^o$, $\alpha_m = 1.01^o$. One *period* of oscillation is defined from $t = 0$ to $t = 2\pi$. In order to compute the entire unsteady flow solution for each period, $\alpha(t)$ is divided into 24 discrete points or *time instances*. Past research [36] has shown that 24 time instances are sufficient to describe the unsteady flow field in an inviscid flow environment.

Unlike traditional optimal control problems, where the control (airfoil shape) is changed along the trajectory, our design approach focuses on creating a single shape whose aerodynamic performance is monitored over a complete pitching cycle.

5.7.1 Full Unsteady Design (Unsteady-Flow Unsteady-Adjoint)

The full unsteady continuous and discrete adjoint based design procedures require the following steps:

1. **Unsteady Flow Calculation**. All numerical simulations are computed for an inviscid flow using a fully implicit second order backward difference formula, a five stage modified Runge-Kutta time stepping scheme is employed at each time instance using a blended first and third order artificial dissipation scheme. A five-level W-cycle multigrid and residual averaging are used to accelerate the convergence. The fully implicit scheme described in sub-section 5.2.1 is then employed to solve for the unsteady flow solution at each time instance. Generally, it requires five periods before a fully periodic solution is achieved. During the last period, the flow solution at each time instance is stored in memory. Fifteen multigrid cycles are used for each time instance. If 24 time instances are used for each cycle and five cycles are used to achieve periodic solution, then a total of 1800 multigrid cycles are required to obtain the unsteady solution.

2. **Perturb α_o to maintain time-averaged lift coefficient**. In order to maintain the time-averaged lift coefficient, the mean angle of attack, α_o, is perturbed. However, α_o, is only modified every three periods, since it requires at least three periods for the global coefficients such as time-averaged lift and drag to converge. A total of 15 periods are needed instead of 5 to achieve the desired time-averaged lift coefficient. This multiplies the total cost by three.

3. **Unsteady Adjoint Calculation**. The unsteady adjoint equation, either the discrete or continuous version, requires integration in reverse time. The same numerical scheme employed to solve the unsteady flow is used here as well with minor adjustments in the code to allow integration in reverse time. Only three periods were needed before a periodic adjoint solution is achieved. 15 multigrid cycles are used for each time instance, which translates to a total of 1080 cycles to achieve a periodic adjoint solution.

4. **Calculate the Gradient**. The expression for the gradient is an integral over time. During the last period of the unsteady adjoint solver, the gradient at each time instance is computed and added to the previous one. At the end of the last period, the complete gradient is available. The gradient is then smoothed using an implicit smoothing technique described in section 3.2.2.

5. **Modify the Airfoil Shape**. The airfoil shape is then modified in the direction of improvement using a steepest descent method.

6. **Update the Grid**. The internal grid is modified based on perturbations on the surface of the airfoil. The method modifies, the grid points along each grid index line projecting from the surface. The arc length between the surface point and the far-field point along the grid line is first computed, then the grid point at each location along the grid line is attenuated proportional to the ratio of its arc length distance from the surface point and the total arc length between the surface and the far-field.

7. **Repeat the Design Process**. The entire design process is repeated until the objective function converges. The problems in this work typically required

between nine to twenty five design cycles to reach the optimum. Each design cycle required 1800 multigrid cycles to compute the flow solution and 1080 cycles for the adjoint solution.

5.7.2 Partial Unsteady Design (Unsteady-Flow Steady-Adjoint)

The design process for the partial unsteady design optimization problem follows that of the full unsteady optimization except for two differences. First, in step 3, instead of an unsteady adjoint computation using the fully implicit scheme, only a steady adjoint computation is performed for all 24 time instances which is equivalent to one period of the oscillation instead of three periods used in the full unsteady optimization case. At each time instance in the computation of the adjoint solution, the respective unsteady flow solution is used to compute the Euler Jacobian matrices. This corresponds to a factor of three computational savings for each design cycle. Thus only 360 multigrid cycles are required. Second, the gradient is no longer a time-averaged of the gradients from each time instance, but rather an average of the gradients from all the time instances.

5.7.3 Time-Averaged-Flow Steady-Adjoint Design

The design process for the time-averaged-flow steady-adjoint design follows that of the partial unsteady design procedure, except for the following two important differences. Here, after the flow solution is complete (periodic flow is achieved), the time-averaged flow solution is computed. Then a single steady adjoint computation is performed using the time-averaged flow solution at the mean angle of attack position, only requiring 15 multigrid cycles. The magnitude of the steady adjoint residuals reduces by 3 orders of magnitude in 15 multigrid cycles. Based on previous work by Nadarajah et al. [60], only a reduction of three orders of magnitude in the adjoint solution is required to obtain accurate gradient values for the Euler equations. The equation for the gradient used in this approach is similar to that used to optimize airfoil shapes in a steady flow environment, however, the flow solution used to calculate the gradient is based upon the time-averaged flow.

5.7.4 Multipoint Design

In the multipoint design approach, the unsteady flow and unsteady adjoint solutions are replaced with steady flow and steady adjoint solutions for each time instance. The gradient is taken as an average of the gradients from each time instance. Since only one period is required for both the flow and adjoint solvers, the total computational cost is 720 multigrid cycles.

Method	Euler	Adjoint	Cost
Multipoint	360	360	1
Time-Averaged	1800	15	2.5
Partial	1800	360	3
Full	1800	1080	4

Table 5.1: Comparison of Computational Cost (Multigrid Cycles) Between Four Design Approaches.

Table 5.1 illustrates a cost comparison between the various design approaches. "Time-Averaged" refers to the time-averaged-flow steady-adjoint design approach, "Partial" refers to the unsteady-flow steady-adjoint approach, and "Full" refers to the unsteady-flow unsteady-adjoint method. Here the middle two columns contain the total number of multigrid cycles used to compute the Euler and adjoint equations. The numbers in the last column represent the ratio of cost of one method with respect to the Multipoint approach. Using the full unsteady design approach requires four times the computational cost of doing the multipoint approach. The difference in cost between one steady Runge-Kutta iteration and one unsteady Runge-Kutta iteration was not factored into the computing cost for each design approach, since the difference is minimal requiring only the addition of the time derivatives of the flow variables for the implicit time stepping for the unsteady algorithm.

5.8 Results

The following sub-sections present results of the time-averaged drag minimization problem for a two-dimensional airfoil undergoing a periodic pitching motion. The first sub-section contains a code and grid validation study. The second sub-section is dedicated to the redesign of the RAE 2822 airfoil to reduce the time-averaged drag coefficient while maintaining the time-averaged lift coefficient. The third section contains results of the time-averaged drag minimization of a VR-7 advanced rotorcraft airfoil. A comparison of a multipoint design, time-averaged-flow steady-adjoint, partial unsteady, and a full unsteady design is explored in the last sub-section.

5.8.1 Code and Grid Validation

Six computational grids are used in the grid validation study. Table 5.2 provides a list of the six different grids. The lens-shape grids are generated using a hyperbolic grid generator. Figure 5.2 illustrates the full 193x33 mesh and a close up view of the NACA 64A010 airfoil. There are 129 points on the surface of the airfoil.

Grid	Dimensions
Coarse 1	193x33
Coarse 2	193x65
Medium 1	257x33
Medium 2	257x65
Fine 1	321x33
Fine 2	321x65

Table 5.2: Euler Lens-Mesh Descriptions

Euler solutions are then computed for each grid and the lift coefficient versus angle of attack is compared with the experimental NACA 64A010 CT6 [15] data. Here the computations are performed at a freestream Mach 0.78, at a mean angle of attack, $\alpha_o = 0^o$ with a maximum angle of attack $\alpha_m = 1.01^o$, and at a reduced frequency, $\omega_r = 0.202$. Five cycles of computation are required in order for the periodic flow to be established, and to allow the time-averaged lift and drag coefficients to converge. Figure 5.3 illustrates the hysteresis loop for all six grids and the CT6 experimental

results. The results are relatively independent of the grid size and reproduce the experimental results with sufficient accuracy. In a steady flow environment, based on linear theory the circulation on the airfoil is constant, however, in a pitching airfoil case, the circulation is a function of time. The wake vorticity shed during the cycle, changes the downwash field over the airfoil. This has the effect of changing the aerodynamic loads on the airfoil and thus the normal and axial coefficients change with time. In figure 5.3, the lift coefficient versus angle of attack loop moves in a counterclockwise direction. The unsteady effect causes an amplitude reduction of the lift coefficient and a phase lag as well. In sub-section 5.8.2 we explore the effect of changing the reduced frequency for the same flow conditions and compare the solutions to steady-state solutions.

The computational result over-estimates the maximum lift coefficient. This is typical of all Euler calculations for unsteady pitching airfoil problems. In viscous flow, the presence of the boundary layer, reduces the fluid velocity in the ξ direction, thus increasing pressure, which leads to a reduction of the pressure difference which in turn reduces the maximum lift coefficient. In this work, the lack of viscous fluxes produces a larger maximum lift coefficient.

Since the primary objective of this work is to formulate the unsteady discrete and continuous inviscid adjoint equations and prove their usefulness, numerical simulations and redesign computations in the upcoming sub-sections will only use the "Coarse 1" grid with 193x33 grid points, since its solutions are within acceptable accuracy.

In figure 5.4 the "Coarse 1", "Medium 1", and "Fine 1" grids, each with 33 points in the η direction, produce similar drag versus angle of attack curves. The drag polar is symmetrical as expected since the airfoil is symmetric. The largest magnitude for the drag coefficient occurs at the largest angle of attack deflection. Increasing the number of points on the airfoil surface does not improve the solution. Increasing the number of points in the η direction, reduces the spatial discretization error in the η direction and improves the accuracy of the calculation. Note here that an increase in the number of points on the surface of the airfoil does not change the solution for the finer mesh that contains 64 points in the η direction.

Figures 5.5 and 5.6 are lift and drag hysteresis loops for the same conditions that are mentioned above but the computations are performed on O-meshes. Increasing the mesh size does not make any appreciable change to the lift versus angle of attack curve but the drag curve does improve in accuracy. As illustrated in the drag polar curves, the 160x32 and the 320x64 mesh have almost the same solution. This is unlike the lens-grid where increasing the number of points in the direction normal to the airfoil surface increased the accuracy of the solution.

Figures 5.7 and 5.10 illustrate the comparison of the lift and drag coefficient versus angle of attack for various o-mesh and lens-grid with the experimental NACA 64A010 CT6 data. The lift hysteresis loop shows no difference between the solutions from the various meshes.

In figure 5.11 we show the convergence history for the steady adjoint, unsteady continuous adjoint, and unsteady discrete adjoint equations. The continuous and discrete unsteady adjoint equations have the same convergence rate. The equations are solved for a RAE 2822 airfoil at Mach 0.78, $\alpha_o = 0^o$, and at a reduced frequency, $\omega_r = 0.202$ on a 193x33 lens-mesh.

5.8.2 RAE 2822: Time-Averaged Drag Minimization with Fixed Time-Averaged Lift Coefficient

Figure 5.12 illustrates the initial and final geometry for the RAE 2822 airfoil. The solid line represents the initial airfoil geometry and the dashed-line illustrates the redesigned airfoil. A distinctive feature of the new airfoil is in the drastic reduction of the upper surface curvature. A reduced curvature leads to a weaker shock and thus a lower wave drag, however, it also leads to a reduction in airfoil camber, resulting to a loss in lift. This effect is desirable for an advancing helicopter rotor blade since it operates at approximately zero lift but undesirable during the retreating phase, since the reduction in camber would reduce the $c_{l_{max}}$ and thus reduce the flight envelope, placing a limit on the forward flight speed.

In order to maintain the time-averaged lift coefficient, TAc_l, the mean angle of attack, α_o, is perturbed to a new value. This results in a need to compute more cycles

to allow the TAc_l and TAc_d to converge. In this work, α_o was perturbed every three cycles. This allowed the TAc_l to converge to a new value before the angle of attack was perturbed any further. A total of 15 complete flow oscillation cycles are used for each design cycle. Figure 5.13 illustrates the convergence rate of the objective function: TAc_d. The TAc_d decreases by 55% from 132 drag counts to 59 drag counts in 11 design cycles. The maximum drag, however, decreases by only 25% from 229 drag counts to 172 drag counts.

In figure 5.14 we show the initial and final lift coefficient hysteresis loops. The results show that to maintain the time-averaged lift coefficient, the mean angle of attack of the new airfoil increased by approximately 1.5 degrees. Also note that the width of the hysteresis loop is almost the same between the two curves. The width of the loop is an indication of the presence of unsteady effects. At very low reduced frequencies, $\omega_r \approx 0$, the lift curve is almost a straight line since very little unsteady effects are present. As the reduced frequency increases, the unsteady effects increase and introduce an amplitude change and a phase lag, thus producing a loop instead of a straight line. Since the widths of the loops are almost identical in width, it can be concluded that the modifications have not altered the slope and the width of the lift curve hysteresis loop. Figure 5.15 illustrates the initial and final drag polar for the RAE airfoil at the same flow condition. The drag polar reflects the unsymmetrical nature of the airfoil. The maximum drag occurs at the largest positive angle of attack and vice versa for the minimum instantaneous drag coefficient. The drag polar for the final design shows a reduction in drag at all angles of attack. A drastic difference between the width of the two curves is illustrated in the figure. The reduction in the loop suggests that the modifications to the surface of the airfoil has altered the airfoil response to the unsteady flow field in such a manner to decrease the amplitude reduction and the phase lag seen in the figure.

Figures 5.16(a-d) illustrate the upper and lower surface instantaneous pressure coefficients for the initial and final designs. In figure 5.16(a), a comparison of the initial instantaneous pressure distribution versus the final at 0^o phase shows an almost complete reduction of the wave drag. The strong shock on the suction side of the airfoil is weakened at all other phases of the oscillation. Another distinctive feature

of the final upper surface pressure distribution at all phases of the pitching motion is the appearance of a flat-top pressure distribution. The reduction of the upper surface curvature is largest between the 30% and 60% chord which contributes towards the flat-top profile. This reduction in the favorable pressure gradient leads to the formation of a weaker shock and thus produces an airfoil with a lower time-averaged drag coefficient. The general trend of the lower surface pressure distribution has only changed by a very small amount at all phases of the pitching motion. This is due mainly to the fact that the magnitude of the gradient along the upper surface is larger especially in the region of the shock wave. The modifications on the lower surface are due mainly to the need to satisfy the thickness constraint rather than the goal of reducing the time-averaged drag coefficient.

Figures 5.17(a-b) show the pressure contours for the initial and final airfoils at 0^o. The sonic line represented by a dashed line is over-plotted on each figure. It is clearly visible that the strong shock on the upper surface of the initial geometry has been almost completely eliminated, although, a supersonic zone still exists. Here the air velocity is accelerated to supersonic speeds and gradually recovers to the freestream pressure at the trailing edge without the need for a discontinuity, such as a shock wave. Another feature of the final design is the location of the sonic transition on the airfoil surface which occurs closer to the leading edge of the airfoil. This is a result of the increase in the airfoil angle of attack which causes the flow to accelerate faster at the leading edge. Similar features are present in the 90^o, 180^o, and 270^o pitching phases as shown in figures 5.18(a-b), 5.19(a-b), and 5.20(a-b).

It is crucial that an optimization technique aimed at improving the performance of airfoils due to an unsteady motion performs well at various different flow conditions to prove its robustness. In this study, the method has been applied to reduce the time-averaged drag coefficient for the following two cases: first, at various reduced frequencies; second, at various freestream Mach numbers. The following are two sub-sections dedicated to presenting results for the above-mentioned cases.

Reduced Frequency Effect

The change in the lift coefficient versus angle of attack for various reduced frequencies is illustrated in figure 5.21. The various solutions are obtained for a freestream Mach number, $M_\infty = 0.78$, at a mean angle of attack, $\alpha_o = 0^o$, and fixed lift coefficient, $C_l = 0.51$. As the reduced frequency is increased, the amplitude reduction continues to decrease but the effect of the phase lag reduces. At a reduced frequency, $\omega_r = 0.45$, the hysteresis loop converges to a straight line and has no phase lag effect. In figure 5.22 the convergence of the time-averaged drag coefficient for various reduced frequencies for the RAE 2822 airfoil is illustrated. At a reduced frequency of 0.050, the time-averaged drag coefficient decreased from 139 drag counts to 80. And at $\omega_r = 0.450$ the reduction is from 132 counts to 57. As the reduced frequency increases the convergence history as plotted in figure 5.22 converges to the same curve. Table 5.3 lists the initial and final time-averaged drag coefficient for various reduced frequencies.

Reduced Frequency	Ini_{TAc_d}	Final_{TAc_d}	Reduction
$\omega_r = 0.050$	139	80	42%
$\omega_r = 0.202$	132	62	53%
$\omega_r = 0.450$	132	57	56%

Table 5.3: Initial and Final Time-Averaged Drag Coefficient for Various Reduced Frequencies using the Full Unsteady Design Approach

Mach Number Effect

In figures 5.23(a-d) and 5.24(a-d) the same design process used for Mach 0.78, is used to design the same RAE 2822 airfoil for two different freestream Mach numbers. Figure 5.23(a-d) illustrates the $M_\infty = 0.76$ flow and figure 5.24(a-d) for the $M_\infty = 0.80$ flow. The final designs follow the same trends as the Mach 0.78 case: first, upper surface curvature reduces, causing a reduction in the favorable pressure gradient, thus reducing the strength of the shock; second, the reduction in the curvature causes a reduction to the lift and thus the airfoil mean angle of attack is increased to control the time-averaged lift coefficient.

These examples show that the full unsteady design optimization is robust, producing a reduction in the time-averaged drag coefficient while maintaining the time-averaged lift coefficient for various freestream velocities and reduced frequencies.

5.8.3 VR-7: Time-Averaged Drag Minimization with Fixed Time-Averaged Lift Coefficient

The VR-7 rotor profile is part of a family of advanced rotor airfoils designed by Leo Dadone [13] at Boeing. The VR-7 (t/c=0.12) and the VR-8 (t/c=0.08) are designed for the rotors on the Heavy lift Helicopter (HLH). The VR-7 profile is used up to the 85% span location of the blade and the VR-8 is used at the tip. According to Dadone, the blade profiles between the 85% and 99% span locations are obtained by interpolating between the VR-7 and VR-8 airfoils [13]. The airfoils are designed to have a very low pitching moment coefficient at zero angle of attack to maintain the oscillatory load level of the control system and high maximum lift coefficient.

Figure 5.25 illustrates the initial and final geometry for the VR-7 airfoil. The solid line represents the initial airfoil geometry and the dashed-line illustrates the redesigned airfoil. Similar to the redesigned RAE 2822, the upper surface curvature of the VR-7 advanced helicopter rotor has reduced as well. Figure 5.26 illustrates the convergence rate of the objective function: TAc_d. The TAc_d reduces by 22% from 309 drag counts to 240 drag counts within 25 design cycles.

In figures 5.27 and 5.28, the initial and final lift and drag hysteresis loops are illustrated. To maintain the time-averaged lift coefficient, the mean angle of attack has increased to $\alpha_o = 2.25^o$. The amplitude and phase lag of the two lift hysteresis loops are almost identical. However, a similar effect seen for the RAE 2822, is present here as well, where the effect of the phase lag reduces for the drag coefficient hysteresis loop. The pressure distribution at four phases of the oscillation is shown in figure 5.29. The final pressure distribution curve shown in red illustrates the small reduction in the strength of the shock which leads to the 22% reduction in the time-averaged drag coefficient.

5.8.4 Comparison of Various Design Approaches

Often a multipoint design approach has been the method of choice for optimization of airfoils in an unsteady flow environment due to its lower computational and memory cost. In this sub-section, we make the argument that even if a multipoint design approach is cheaper, it cannot replace a full unsteady optimization. The following results will show that there are benefits to unsteady optimization.

In order to compare the multipoint design approach to the full unsteady optimization, several unsteady design cases are tested at various reduced frequencies. It is expected that at very low reduced frequencies, the flow characteristics are very similar to those of steady-state computations. Full unsteady design cases are computed at reduced frequencies ranging from 0.050 to 0.450 at a Mach 0.78. The time-averaged lift coefficient for all reduced frequencies are fixed to the same value of $TAc_l = 0.51$.

Figure 5.30 illustrates a comparison of the lift coefficient hysteresis loops between the steady and unsteady cases at various reduced frequencies. The plots illustrate the impact of the unsteady effects on the reduction in amplitude and phase lag.

Figure 5.31 illustrates a comparison of the various airfoil geometries including the initial airfoil, airfoils designed at various reduced frequencies, and airfoils designed using the multipoint approach. The airfoils are designed at Mach 0.78, at a mean angle of attack, $\alpha_o = 0^o$, and an angle of attack deviation of $\alpha_m = \pm 1.01^o$. The difference in the final airfoil geometry between designs performed at various reduced frequencies is very small except in areas on the upper surface where a greater reduction in the curvature is seen for higher reduced frequencies.

In figure 5.31 the airfoil designed using the multipoint approach is almost identical to the one designed using a full unsteady optimization approach at a reduced frequency of $\omega_r = 0.050$. Figure 5.32 further supports this fact with an identical, to within the numerical accuracy of the code, final time-averaged drag coefficient of 80 drag counts. This result verifies the solution from the full unsteady optimization problem, since this was an expected answer. Table 5.4 shows a comparison between the multipoint design and a full unsteady optimization computed at a reduced frequency of 0.050. The time-averaged lift coefficient is constrained at 0.51. In the multipoint case, instead of the time-averaged lift, the average lift from the 24 time

instances are used as a constraint. Since the lift constraint from each method is calculated differently, the comparison between the two approaches is not perfect but we believe that the difference is very small since the multipoint design is based upon all 24 time instances. Anything less would further increase the error between the two comparisons.

Case	Ini_{TAc_d}	Final_{TAc_d}	Reduction
Full Unsteady, $\omega_r = 0.050$	139	80	42%
Multipoint	146	82	43%

Table 5.4: Comparison Between the Multipoint and Full Unsteady Optimization

Next the time-averaged-flow steady-adjoint and the partial unsteady design approaches are used to optimize the RAE 2822 airfoil at the flow conditions described above. Since the design computation calculated at a reduced frequency of 0.450 using the full unsteady optimization approach produces the airfoil with the lowest time-averaged drag coefficient, then this is the best case to be compared with the time-averaged-flow steady-adjoint and partial unsteady design approaches. Figure 5.32 illustrates that the history of the time-averaged drag coefficient is almost identical between the three design approaches. Table 5.5 contains a comparison of the initial and final time-averaged drag coefficients for the full unsteady, partial unsteady, and time-averaged-flow steady-adjoint approaches. The results vary by at most two drag counts at each design cycle. This result demonstrates that the time-averaged-flow steady-adjoint approach for the reduction of the time-averaged drag coefficient of a pitching airfoil in transonic flow using the Euler equations is sufficient. However, this result is only for a particular set of problems. The issue of what type of problems will require a full unsteady optimization technique remains open.

Case	Ini_{TAc_d}	Final_{TAc_d}	Reduction
Full Unsteady, $\omega_r = 0.450$	132	57	56%
Partial $\text{Unst.}_{0.450}$	132	58	56%
$\text{TA-Flow Steady}_{0.450}$	132	58	56%

Table 5.5: Initial and Final Time-Averaged Drag Coefficient for Various Design Approaches

In figure 5.33 the gradients computed using the full continuous and discrete unsteady adjoint methods are compared to gradients computed using the time-averaged-flow steady-adjoint and partial unsteady approaches. These gradients are computed after the first design cycle. There is an identical match between the full continuous and discrete unsteady adjoint methods. The time-averaged-flow steady-adjoint gradients agree very well with the full and partial unsteady adjoint gradients. The sharp peak at grid point 115 illustrates the large gradient values close to the location of the shock wave. At this grid location, the partial unsteady and time-averaged-flow steady-adjoint gradients have better agreement.

The next set of numerical evaluations compare the final designs of the full unsteady and the multipoint design approaches. Full unsteady simulations are computed at three reduced frequencies with the airfoil designed using the multipoint technique. Table 5.6 shows the time-averaged drag coefficient for various reduced frequencies for the full unsteady and multipoint design. The information in the table provides the following information. First, the time-averaged drag coefficient at a reduced frequency of $\omega_r = 0.050$ for the multipoint design airfoil is very similar to the average drag coefficient obtained as the final design point from table 5.4. This confirms that at low reduced frequencies, a multipoint design is sufficient, and provides final designs that are comparable to those obtained from the full unsteady design approach. This is further confirmed in the second column of table 5.6. Second, as the reduced frequency is increased, the time-averaged drag coefficient of the multipoint design reduces further. However, the benefit of a full unsteady approach over the multipoint case begins to increase. At a reduced frequency of $\omega_r = 0.450$, the gain is 9%.

Case	$\omega_r = 0.050$	$\omega_r = 0.202$	$\omega_r = 0.050$
Full Unsteady	80	62	57
Multipoint	83	70	68
Benefit of Full Unsteady over Multipoint Design Approach	2%	6%	9%

Table 5.6: Time-Averaged Drag Coefficient for Various Reduced Frequencies for the Full Unsteady and Multipoint Design

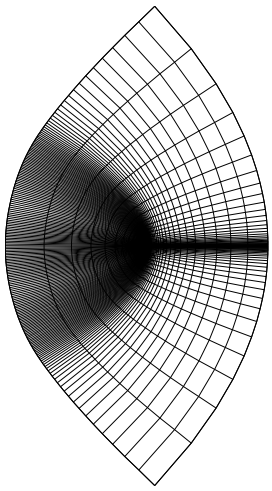


Figure 5.1: Lens-Mesh 192x32: NACA 64A010

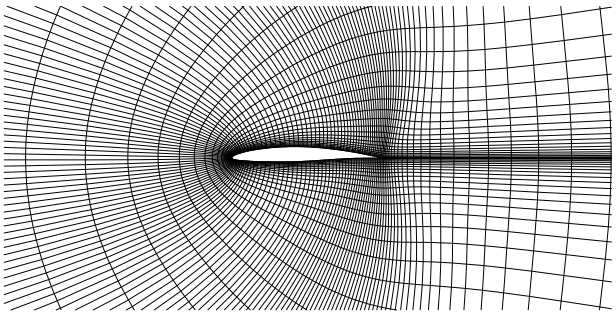


Figure 5.2: Close-up View: Lens-Mesh 192x32: NACA 64A010

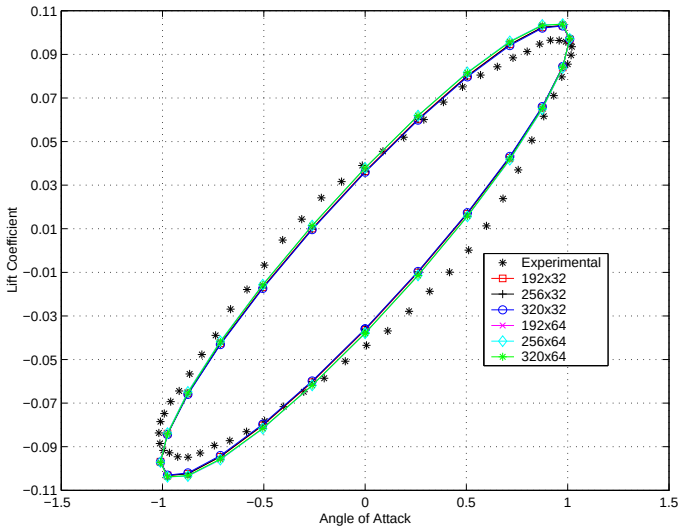


Figure 5.3: Comparison of Lift Coefficient versus Angle of Attack for Various Lenz-Mesh Grids and Experimental Results on a NACA 64A010 CT6 Case.

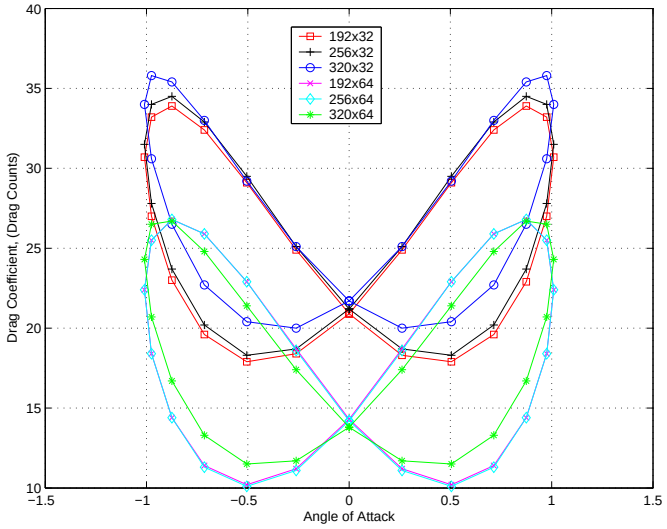


Figure 5.4: Comparison of Drag Coefficient versus Angle of Attack for Various Lenz-Mesh Grids on a NACA 64A010 CT6 Case.

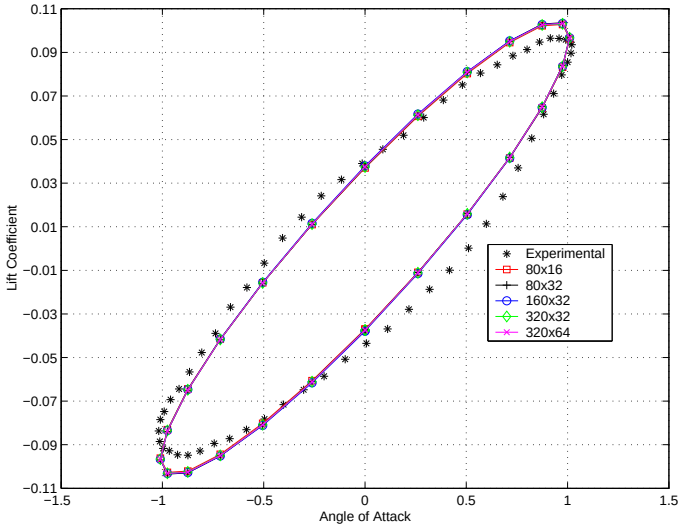


Figure 5.5: Comparison of Lift Coefficient versus Angle of Attack for Various O-Mesh Grids and Experimental Results on a NACA 64A010 CT6 Case.

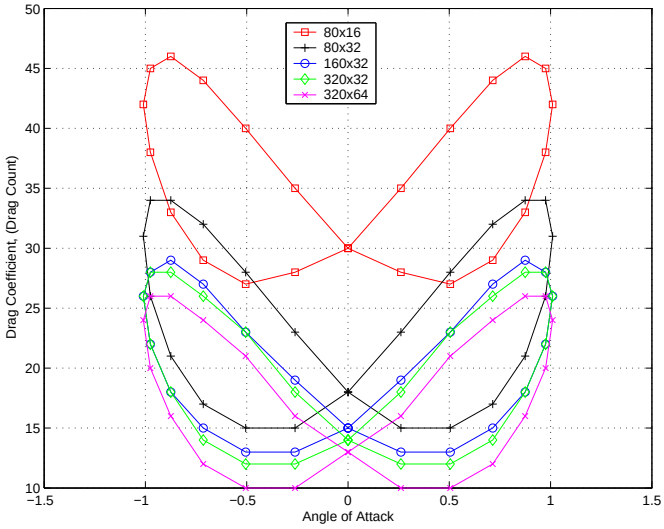


Figure 5.6: Comparison of Drag Coefficient versus Angle of Attack for Various O-Mesh Grids on a NACA 64A010 CT6 Case.

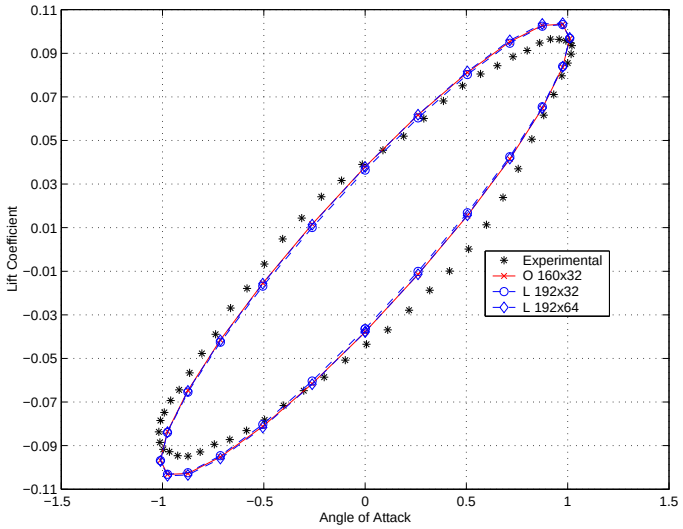


Figure 5.7: Comparison of Lift Coefficient versus Angle of Attack for Various O-Mesh, Lenz-Mesh Grids and Experimental Results on a NACA 64A010 CT6 Case.

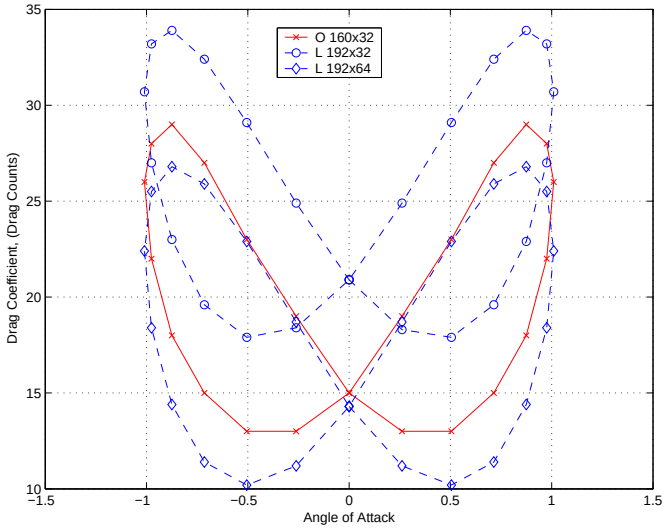


Figure 5.8: Comparison of Drag Coefficient versus Angle of Attack for Various O-Mesh, Lenz-Mesh Grids and Experimental Results on a NACA 64A010 CT6 Case.

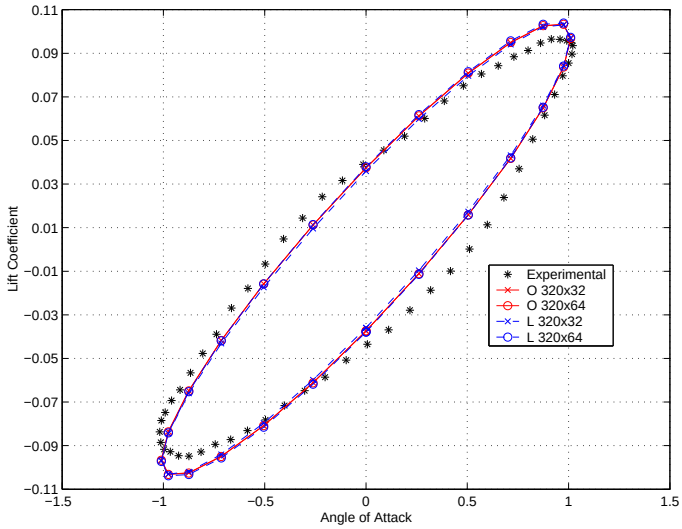


Figure 5.9: Comparison of Lift Coefficient versus Angle of Attack for Various O-Mesh, Lenz-Mesh Grids and Experimental Results on a NACA 64A010 CT6 Case.

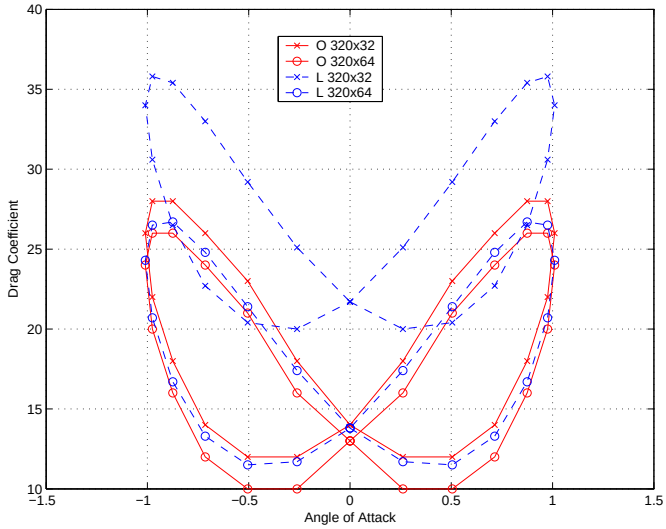


Figure 5.10: Comparison of Drag Coefficient versus Angle of Attack for Various O-Mesh, Lenz-Mesh Grids and Experimental Results on a NACA 64A010 CT6 Case.

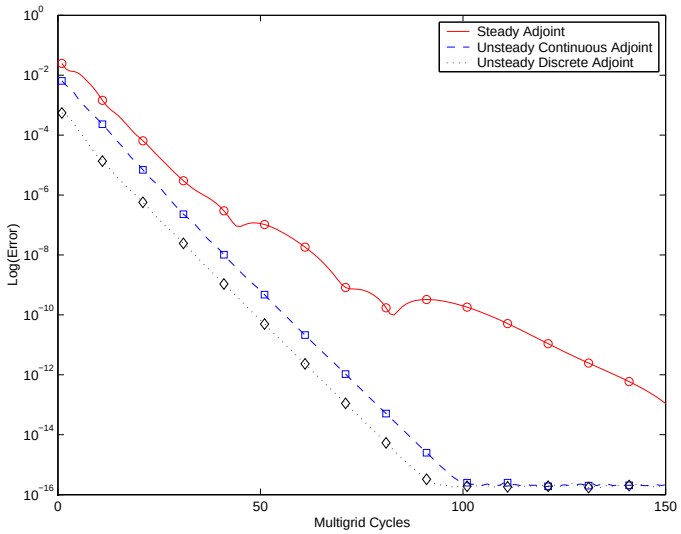


Figure 5.11: Convergence History of the Steady, Unsteady Continuous, and Unsteady Discrete Adjoint Equations. 193x33 Lens-Mesh. RAE 2822 Airfoil, $M_\infty = 0.78$, $\omega_r = 0.202$, $\alpha_o = 0^o$

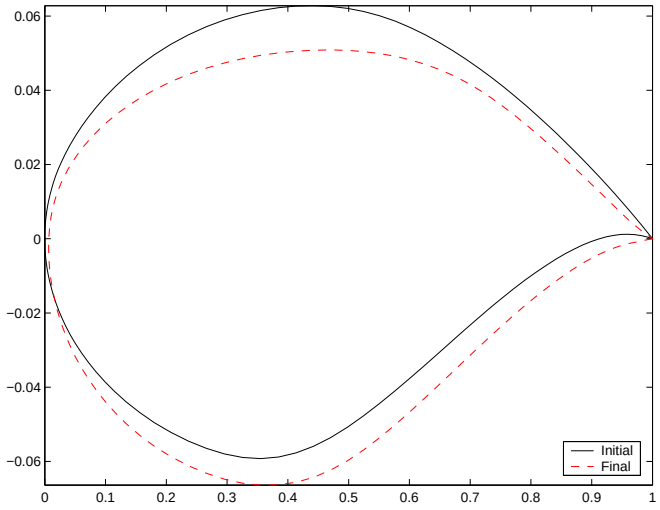


Figure 5.12: Initial and Final Geometry for a RAE 2822 Airfoil at $M_\infty = 0.78$, $\omega_r = 0.202$, $\bar{\alpha} = 0^o$

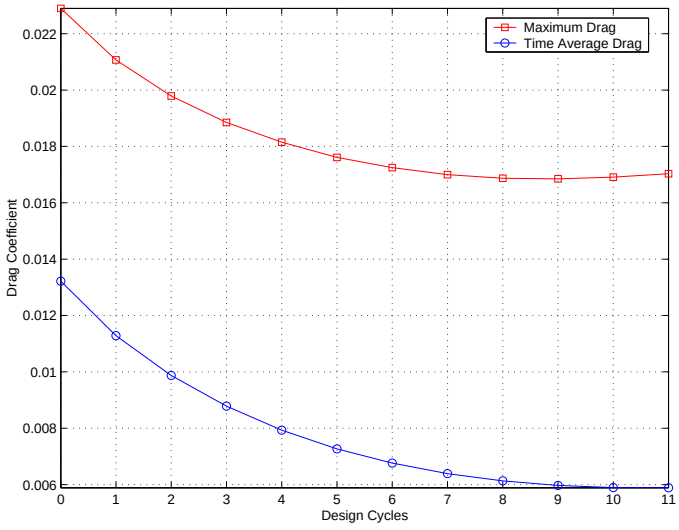


Figure 5.13: Convergence of the Maximum and Time-Averaged Drag Coefficients for the RAE 2822 a $M_\infty = 0.78$, $\omega_r = 0.202$, $\bar{\alpha} = 0^o$

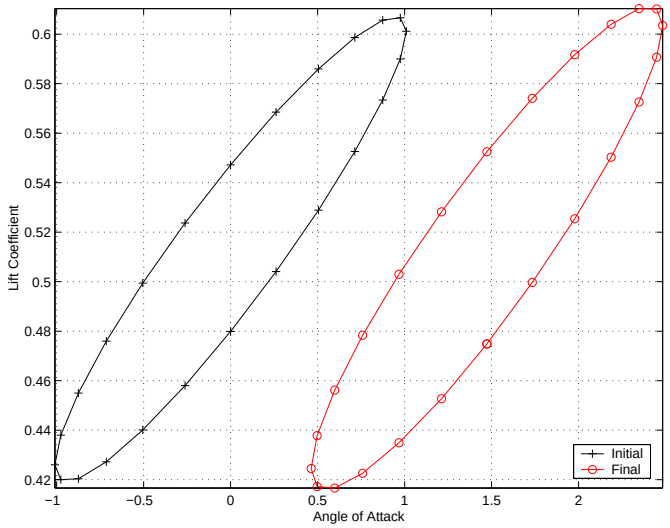


Figure 5.14: Initial and Final Lift Coefficient Versus Angle of Attack for a RAE 2822 Airfoil at $M_\infty = 0.78$, $\omega_r = 0.202$, $\bar{\alpha} = 0^o$

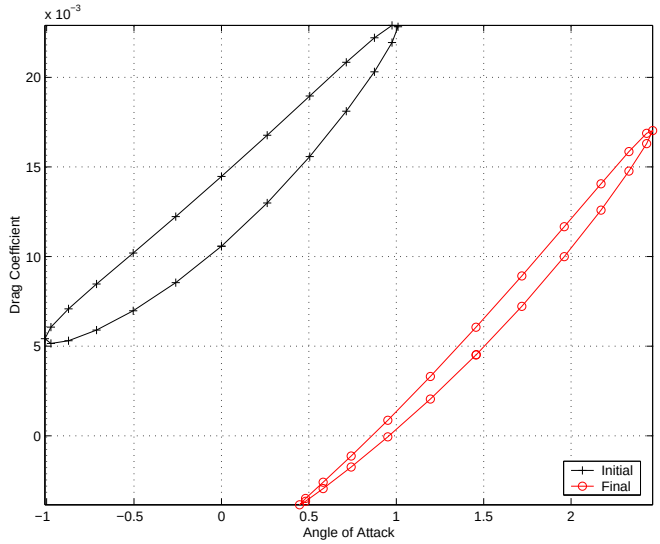
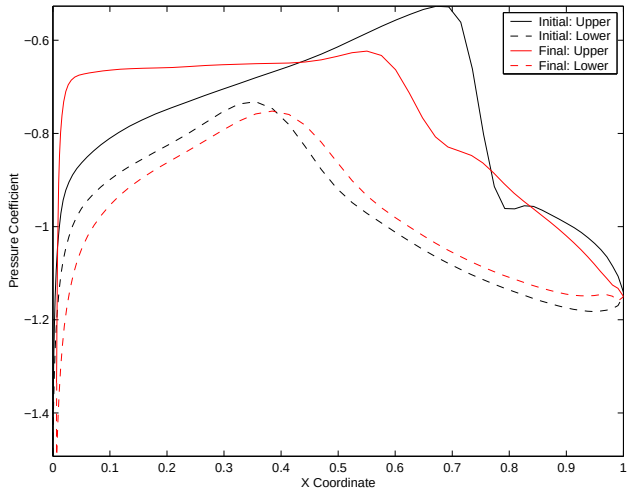
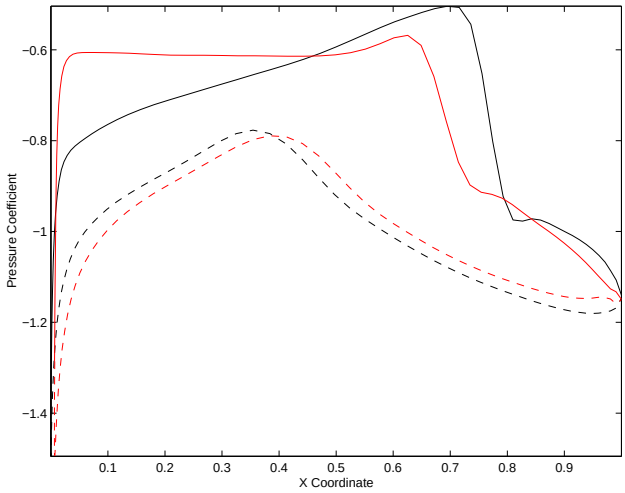


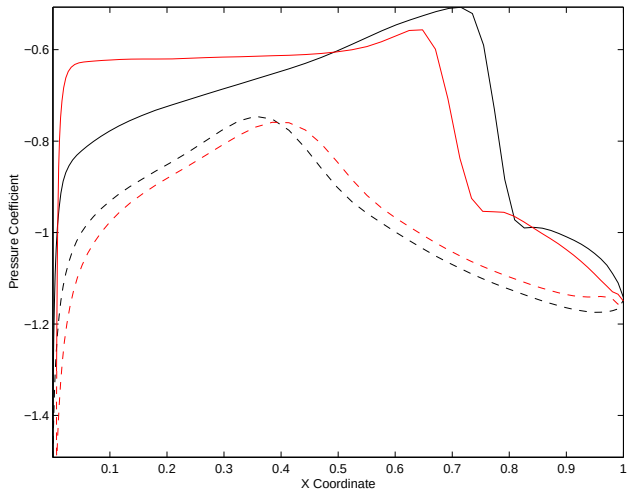
Figure 5.15: Initial and Final Drag Coefficient Versus Angle of Attack for a RAE 2822 Airfoil at $M_\infty = 0.78$, $\omega_r = 0.202$, $\bar{\alpha} = 0^o$



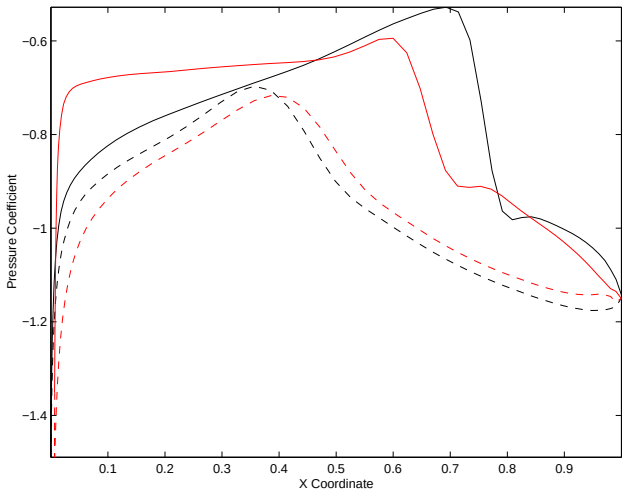
16a: Phase $= 0^o$



16b: Phase $= 90^o$

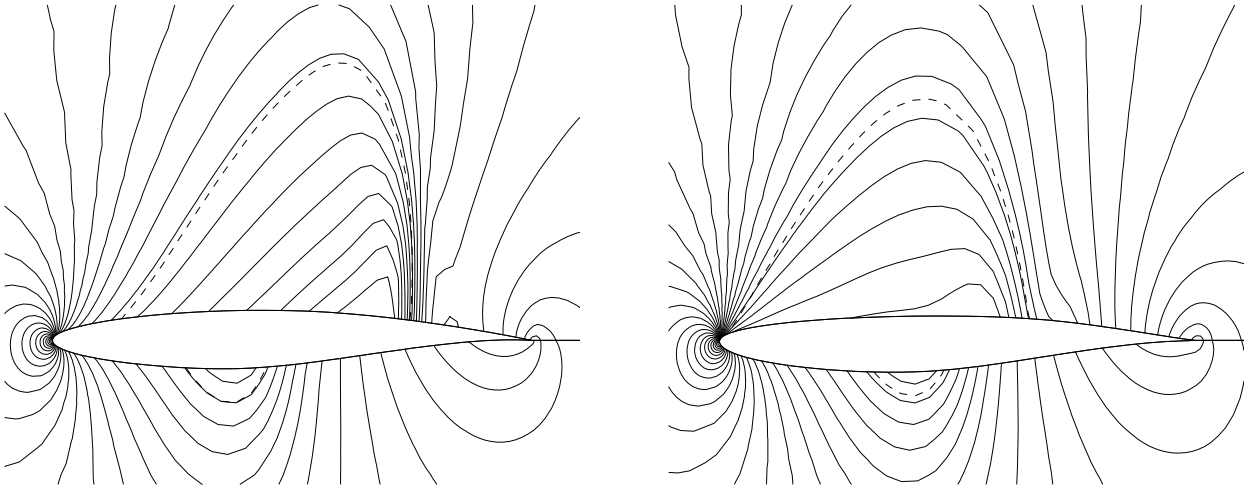


16c: Phase $= 180^o$



16d: Phase $= 270^o$

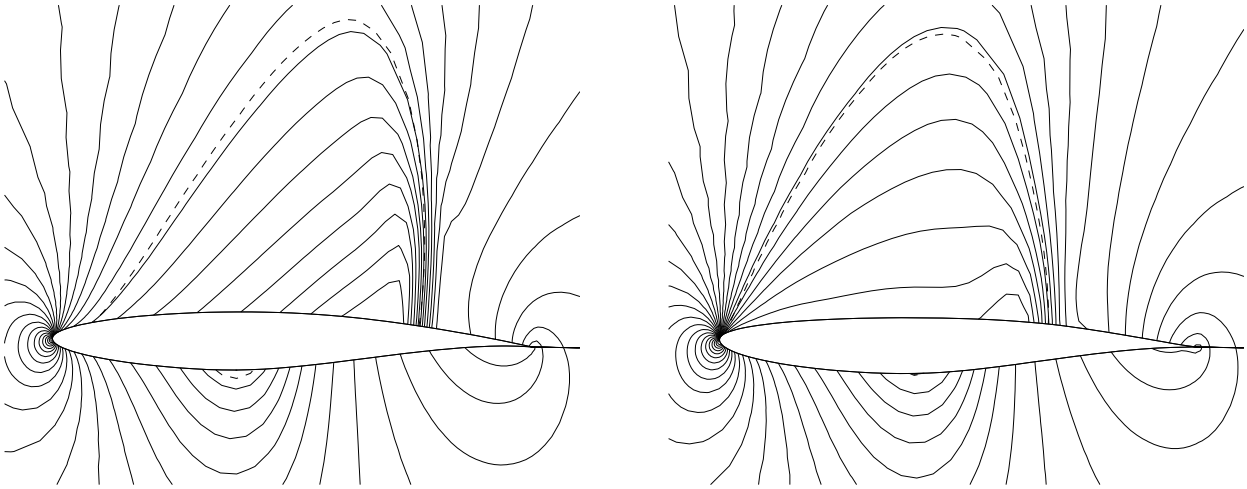
Figure 5.16: Initial and Final Pressure Coefficients at Various Phases for a RAE 2822 Airfoil at $M_\infty = 0.78$, $\omega_r = 0.202$, $\bar{\alpha} = 0^o$



17a: Initial Airfoil

17b: Final Design

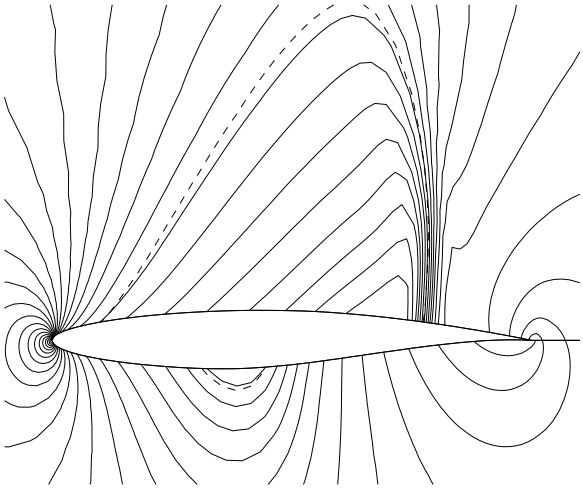
Figure 5.17: Pressure Contour Plot for RAE 2822 Airfoil at Phase = 0^o
Grid - 192 x 32, $M_\infty = 0.78$, $\omega_r = 0.202$, Fixed $C_l = 0.534$



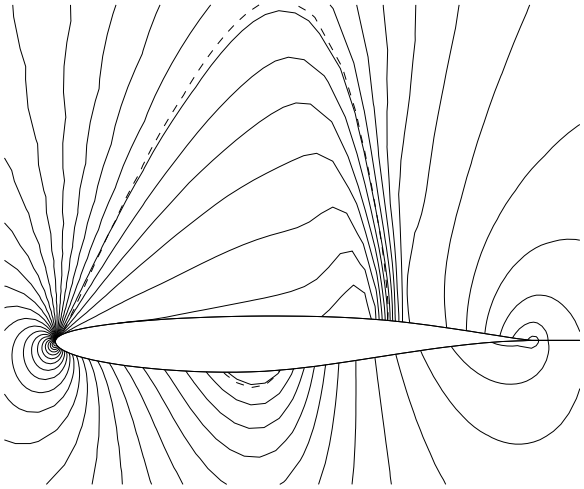
18a: Initial Airfoil

18b: Final Design

Figure 5.18: Pressure Contour Plot for RAE 2822 Airfoil at Phase = 90^o
Grid - 192 x 32, $M_\infty = 0.78$, $\omega_r = 0.202$, Fixed $C_l = 0.534$

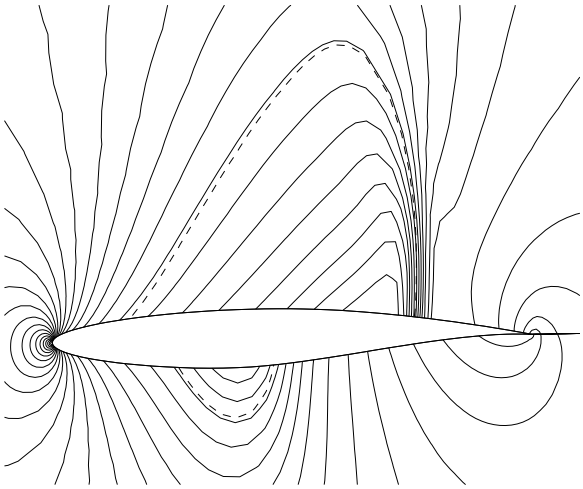


19a: Initial Airfoil

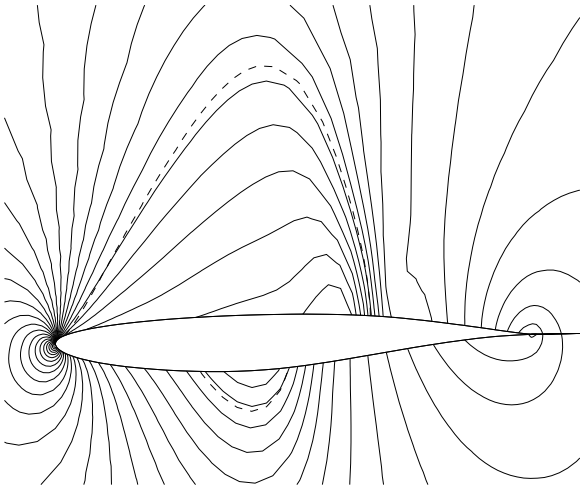


19b: Final Design

Figure 5.19: Pressure Contour Plot for RAE 2822 Airfoil at Phase = 180^o
Grid - 192 x 32, $M_\infty = 0.78$, $\omega_r = 0.202$, Fixed $C_l = 0.534$



20a: Initial Airfoil



20b: Final Design

Figure 5.20: Pressure Contour Plot for RAE 2822 Airfoil at Phase = 270^o
Grid - 192 x 32, $M_\infty = 0.78$, $\omega_r = 0.202$, Fixed $C_l = 0.534$

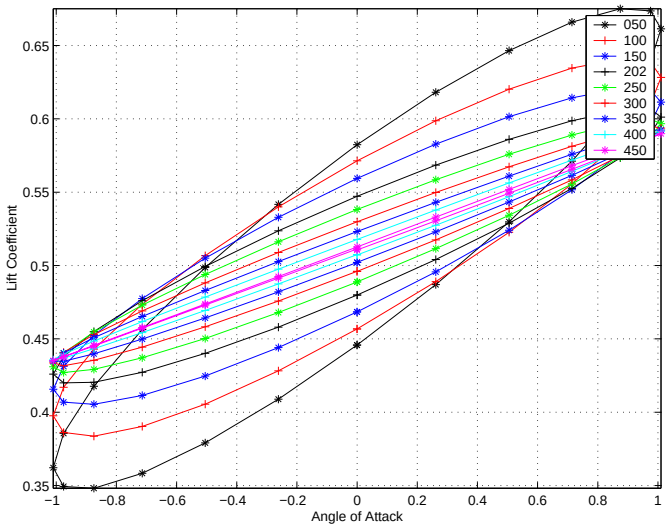


Figure 5.21: Lift Coefficient Versus Angle of Attack for Various Reduce Frequencies for the RAE 2822 Airfoil at $M_\infty = 0.78$, $\bar{\alpha} = 0^o$, Fixed $C_l = 0.51$

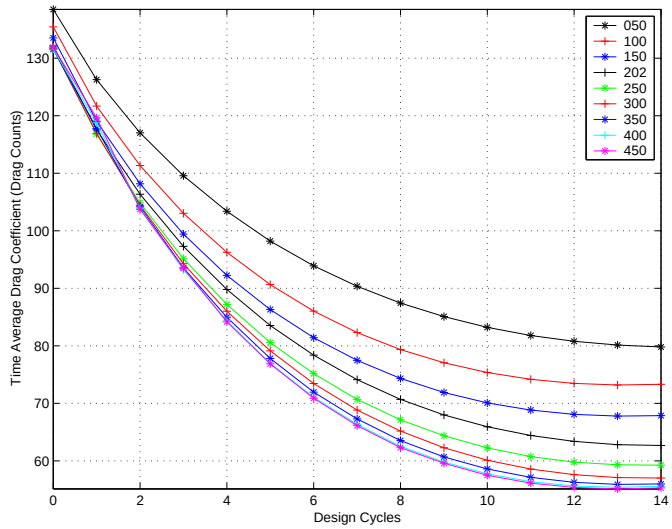
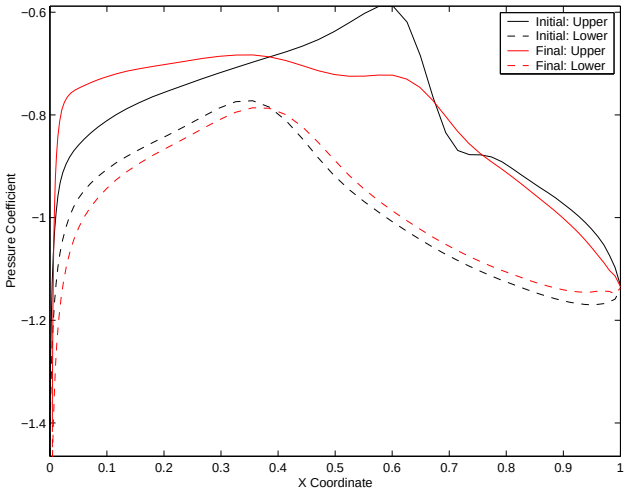
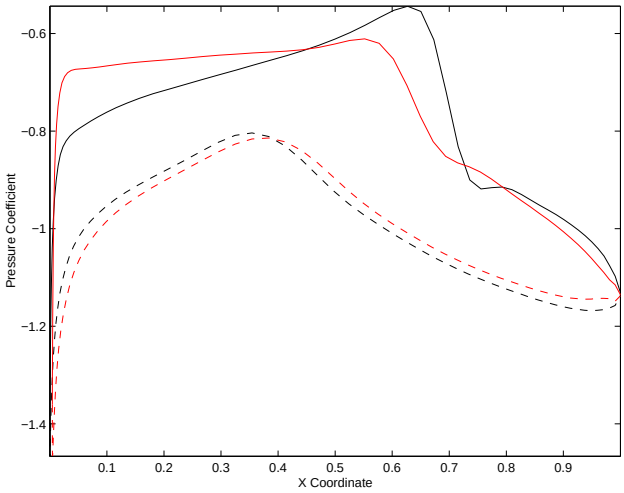


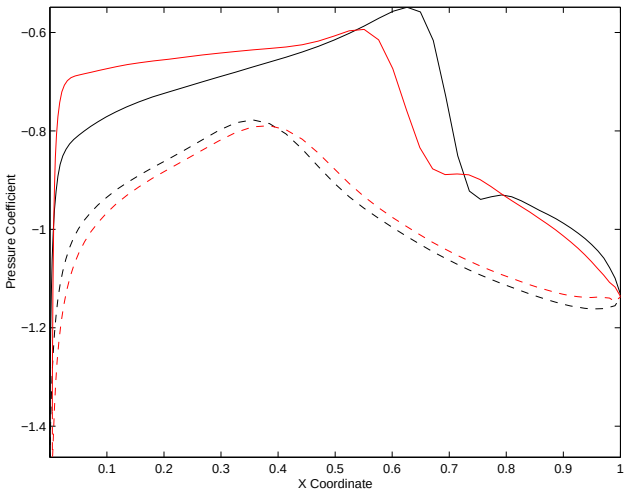
Figure 5.22: Convergence of the Time-Averaged Drag Coefficient for Various Reduced Frequencies for the RAE 2822 Airfoil at $M_\infty = 0.78$, $\bar{\alpha} = 0^o$, Fixed $C_l = 0.51$



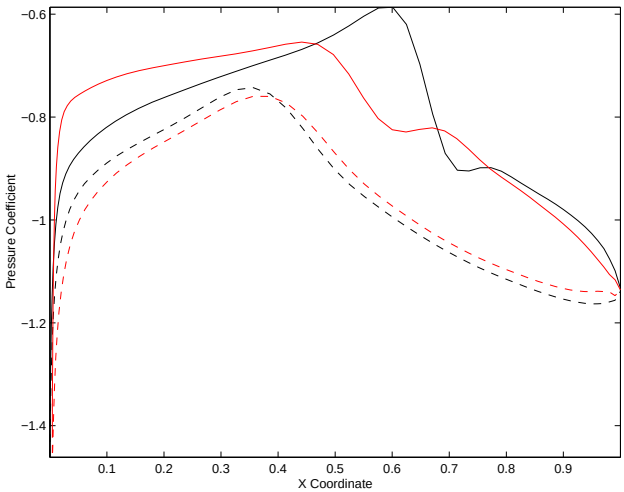
23a: Phase $= 0^o$



23b: Phase $= 90^o$

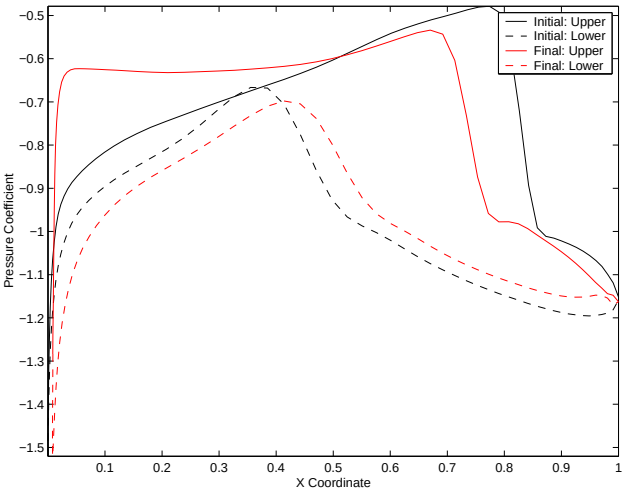


23c: Phase $= 180^o$

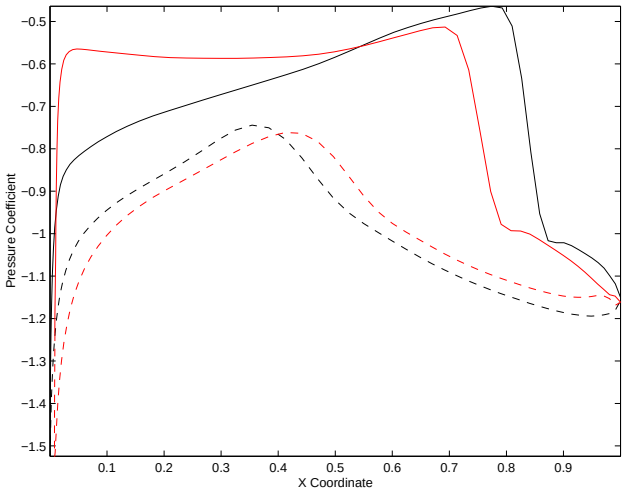


23d: Phase $= 270^o$

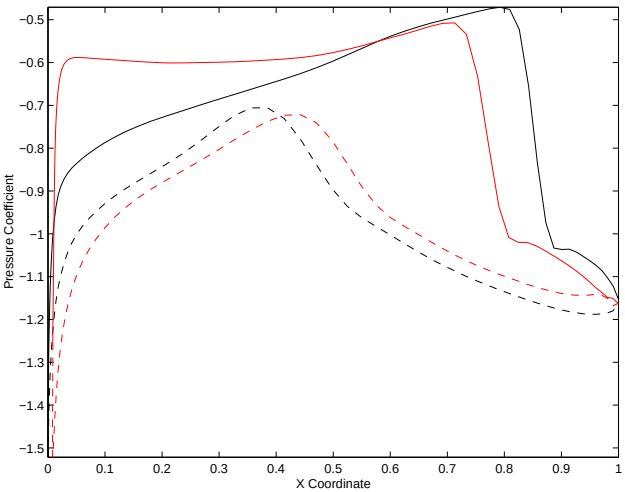
Figure 5.23: Initial and Final Pressure Coefficients at Various Phases for a RAE 2822 Airfoil at $M_\infty = 0.76$, $\omega_r = 0.202$, $\bar{\alpha} = 0^o$



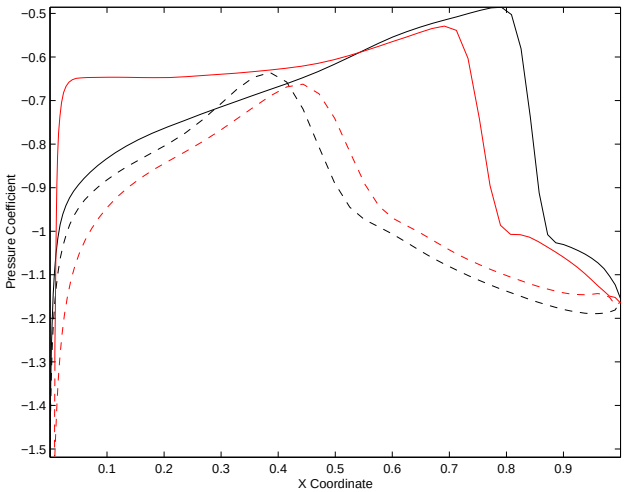
24a: Phase $= 0^o$



24b: Phase $= 90^o$



24c: Phase $= 180^o$



24d: Phase $= 270^o$

Figure 5.24: Initial and Final Pressure Coefficients at Various Phases for a RAE 2822 Airfoil at $M_\infty = 0.80$, $\omega_r = 0.202$, $\bar{\alpha} = 0^o$

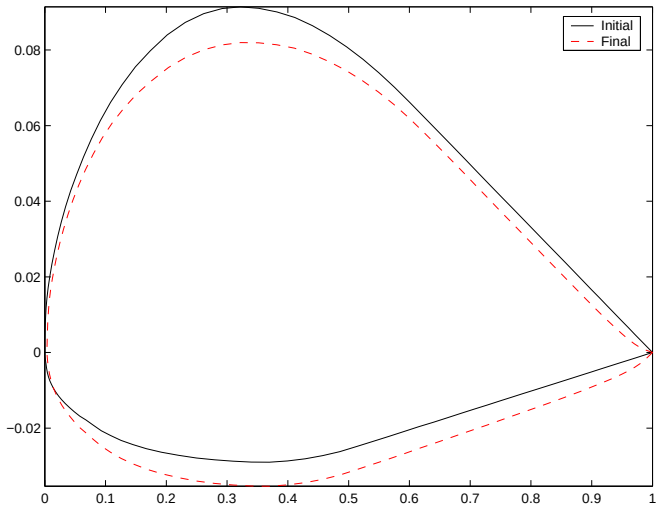


Figure 5.25: Initial and Final Geometry for a VR-7 Airfoil at $M_\infty = 0.75$, $\omega_r = 0.202$, $\bar{\alpha} = 0^o$

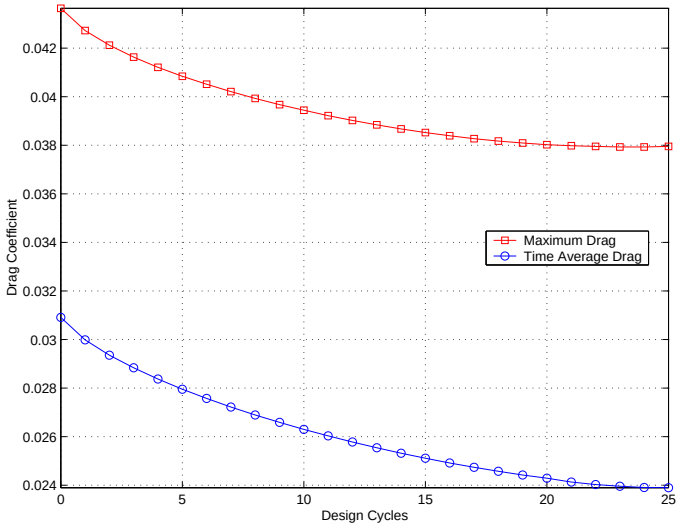


Figure 5.26: Convergence of the Maximum and Time-Averaged Drag Coefficients for the VR-7 a $M_\infty = 0.75$, $\omega_r = 0.202$, $\bar{\alpha} = 0^o$

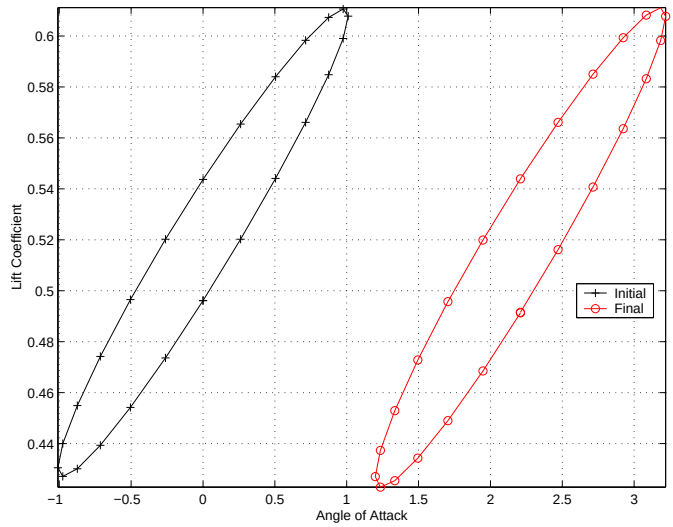


Figure 5.27: Initial and Final Lift Coefficient Versus Angle of Attack for a VR-7 Airfoil at $M_\infty = 0.75$, $\omega_r = 0.202$, $\bar{\alpha} = 0^o$

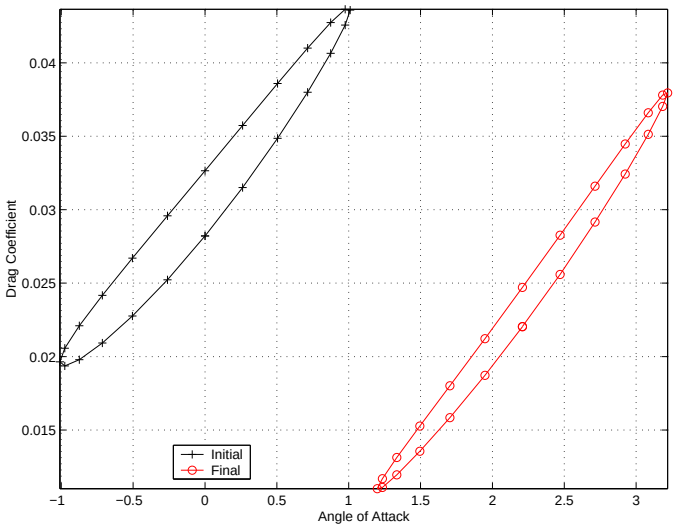
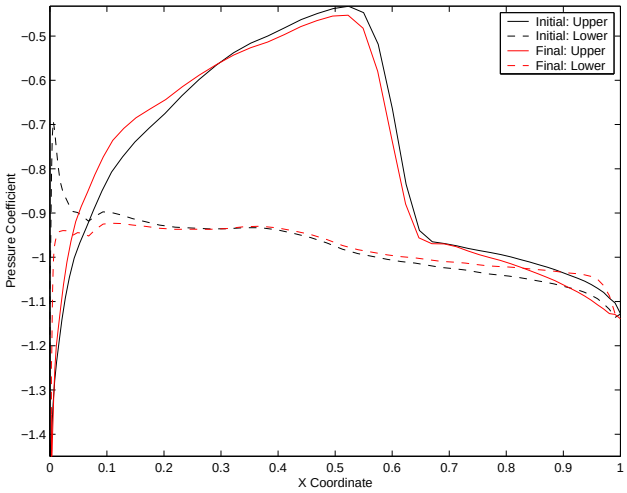
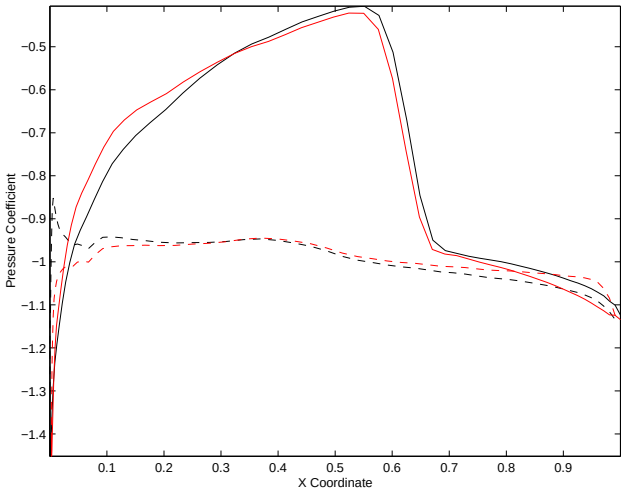


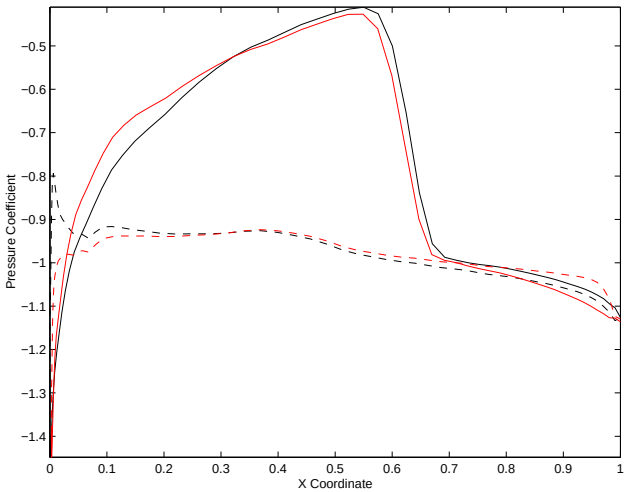
Figure 5.28: Initial and Final Drag Coefficient Versus Angle of Attack for a VR-7 Airfoil at $M_\infty = 0.75$, $\omega_r = 0.202$, $\bar{\alpha} = 0^o$



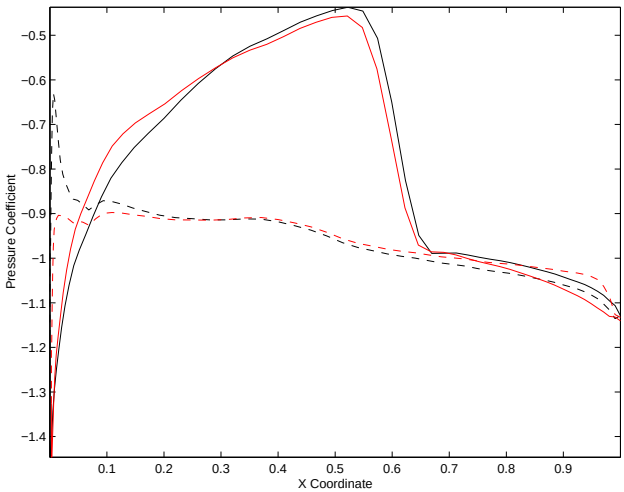
29a: Phase $= 0^o$



29b: Phase $= 90^o$



29c: Phase $= 180^o$



29d: Phase $= 270^o$

Figure 5.29: Initial and Final Pressure Coefficients at Various Phases for a VR-7 Airfoil at $M_\infty = 0.75$, $\omega_r = 0.202$, $\bar{\alpha} = 0^o$

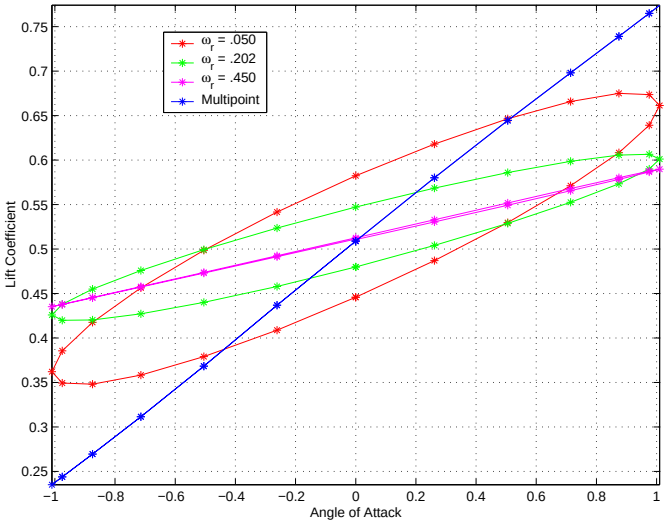


Figure 5.30: A Comparison of the Lift Coefficient Versus Angle of Attack for Various Reduced Frequencies and the Multipoint Approach for the RAE 2822 Airfoil at $M_\infty = 0.78$, $\bar{\alpha} = 0^o$,Fixed $C_l = 0.51$

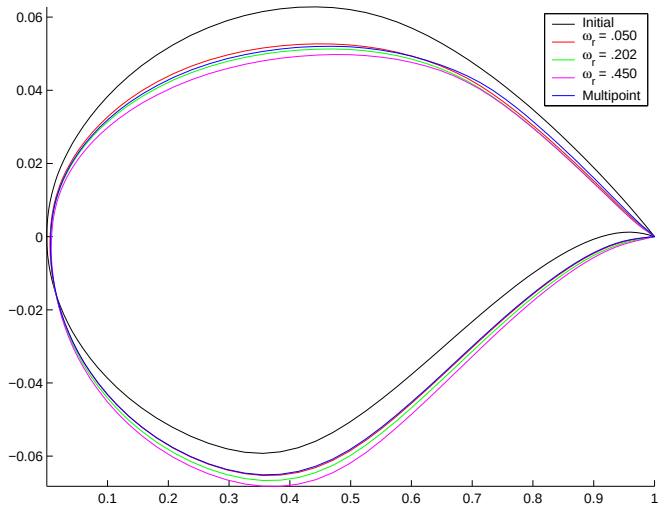


Figure 5.31: A Comparison of Final Airfoil Geometries Between the Initial Airfoil, Airfoils Designed with Various Reduced Frequencies, and Airfoil Designed using the Multipoint Approach. $M_\infty = 0.78$, $\bar{\alpha} = 0^o$, Fixed $C_l = 0.51$

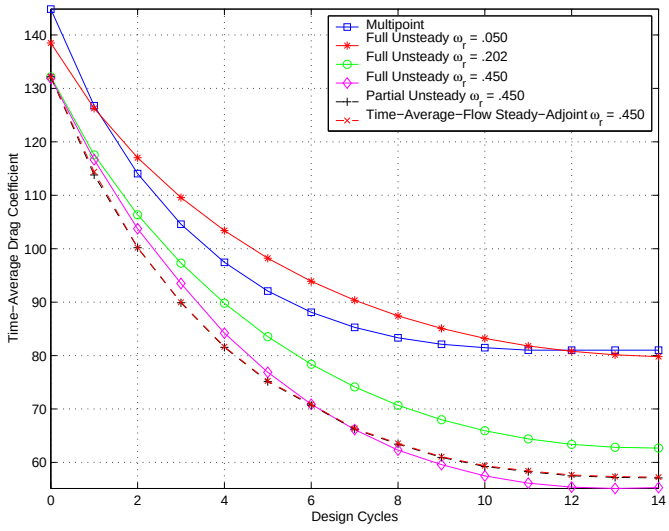


Figure 5.32: A Comparison of the Convergence of the Time-Averaged Drag Coefficient for Various Reduced Frequencies and the Multipoint Approach for the RAE 2822 Airfoil at $M_\infty = 0.78$, $\bar{\alpha} = 0^o$, Fixed $C_l = 0.51$

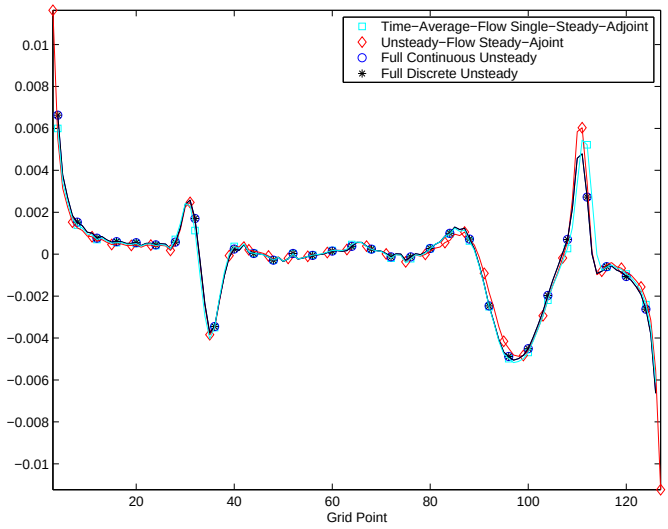


Figure 5.33: Comparison of Time-Averaged-Flow Steady-Adjoint, Partial Unsteady, Full Continuous and Discrete Unsteady Gradients. RAE 2822 Airfoil, $M_\infty = 0.78$, $\alpha_o = 0^o$, Fixed $C_l = 0.51$

Chapter 6

The Remote Inverse Design Problem

This chapter presents an adjoint method for the calculation of remote sensitivities in supersonic flow. The goal is to quantify the influence of geometry modifications on the pressure distribution at an arbitrary location in the near field of the domain of interest. First, the complete formulation and discretization of the continuous and discrete adjoint equations are derived. The special treatment of the adjoint boundary condition to obtain remote sensitivities is also discussed. Second, we present results that demonstrate the application of the theory to a three-dimensional remote inverse design problem using both a low sweep biconvex wing and a highly swept blunt leading edge wing. Third, we demonstrate results that establish the added benefit of using an objective function that contains the sum of the remote inverse and drag minimization cost functions.

6.1 The Remote Inverse Design Problem

Our approach to the remote inverse design problem is similar to the formulation of the adjoint equations discussed in Chapters 4 and 5. However, in this approach, an arbitrary line integral is introduced into the computational domain to allow the calculation of remote sensitivities.

6.1.1 Formulation of the Continuous Adjoint Equations for the Remote Inverse Problem

The aerodynamic properties that define the cost function are functions of the flow field variables, w, and the physical location of the boundary, which may be represented by the function S.

Suppose that the performance is measured by a cost function

$$I = \varpi_1 \int_{\mathcal{B}_W} \mathcal{M}\left(w, S\right) d\mathcal{B}_\xi + \varpi_2 \int_{\mathcal{B}_{NF}} \mathcal{N}\left(w, S\right) d\mathcal{B}_\xi, \tag{6.1}$$

containing both wall boundary ($\mathcal{B}_W$) and near field boundary ($\mathcal{B}_{NF}$) contributions, where $d\mathcal{B}_\xi$ includes the surface and near field elements in the computational domain, while ϖ_1 and ϖ_2 are the weighting coefficients. The coordinates ξ_i that describe the fixed computational domain are chosen so that each boundary conforms to a constant value of one of these coordinates. In general, $\mathcal{M}$ and $\mathcal{N}$ will depend on both the flow variables w and the metrics S defining the computational space.

The design problem is now treated as a control problem where the boundary shape represents the control function, which is chosen to minimize I subject to the constraints defined by the flow equations. A shape change produces a variation in the flow solution, δw, and the metrics, δS, which in turn produce a variation in the cost function

$$\delta I = \varpi_1 \int_{\mathcal{B}_W} \delta\mathcal{M}(w, S)\, d\mathcal{B}_\xi + \varpi_2 \int_{\mathcal{B}_{NF}} \delta\mathcal{N}(w, S)\, d\mathcal{B}_\xi, \tag{6.2}$$

with

$$\begin{aligned} \delta\mathcal{M} &= \left[\mathcal{M}_w\right]_I \delta w + \delta\mathcal{M}_{II}, \\ \delta\mathcal{N} &= \left[\mathcal{N}_w\right]_I \delta w + \delta\mathcal{N}_{II}, \end{aligned} \tag{6.3}$$

where we use the subscripts I and II to distinguish between the contributions associated with the variation of the flow solution δw and those associated with the metric variations δS. Thus $\left[\mathcal{M}_w\right]_I$ and $\left[\mathcal{N}_w\right]_I$ represent $\frac{\partial \mathcal{M}}{\partial w}$ and $\frac{\partial \mathcal{N}}{\partial w}$ with the metrics fixed,

while $\delta\mathcal{M}_{II}$ and $\delta\mathcal{N}_{II}$ represent the contribution of the metric variations δS to $\delta\mathcal{M}$ and $\delta\mathcal{N}$ with the flow solution fixed. The weak form of the Euler equations for steady flow is

$$\int_{\mathcal{D}} \frac{\partial \psi^T}{\partial \xi_i} \delta F_i d\mathcal{D} = \int_{\mathcal{B}} n_i \psi^T \delta F_i d\mathcal{B},$$

where the test vector ψ is an arbitrary differentiable function, and n_i is the outward normal at the boundary. If a differentiable solution, w, for this equation is obtained, then it can be integrated by parts to give

$$\int_{\mathcal{D}} \psi^T \frac{\partial}{\partial \xi_i} \delta F_i d\mathcal{D} = 0. \tag{6.4}$$

Since this is true for any ψ, the differential form can be recovered. Here δF_i can be split into contributions associated with δw and δS using a similar notation

$$\delta F_i = [F_{iw}]_I \, \delta w + \delta F_{iII}, \quad \text{where} \quad [F_{iw}]_I = S_{ij} \frac{\partial f_j}{\partial w}.$$

The domain can then be split into two parts as shown in figure 6.1. First, the near field domain ($\mathcal{D}_1$) whose boundaries are the wing surface and the near field boundary plane. Second, the far-field domain ($\mathcal{D}_2$) which borders the near field domain along the near field boundary plane and the far-field boundary. Thus equation (6.4) can be written as

$$\int_{\mathcal{D}_1} \psi^T \frac{\partial}{\partial \xi_i} \delta F_i d\mathcal{D}_\xi + \int_{\mathcal{D}_2} \psi^T \frac{\partial}{\partial \xi_i} \delta F_i d\mathcal{D}_\xi = 0.$$

This may be integrated by parts to give

$$\begin{aligned} &\int_{\mathcal{B}_W} n_i \psi^T \delta F_i d\mathcal{B}_\xi - \int_{\mathcal{D}_1} \frac{\partial \psi^T}{\partial \xi_i} \delta F_i d\mathcal{D}_\xi \\ + &\int_{\mathcal{B}_{NF}} n_i \left(\psi^+ - \psi^-\right)^T \delta F_i d\mathcal{B}_\xi - \int_{\mathcal{D}_2} \frac{\partial \psi^T}{\partial \xi_i} \delta F_i d\mathcal{D}_\xi = 0 \end{aligned} \tag{6.5}$$

where ψ^+ and ψ^- are the values of the Lagrange Multiplier, ψ, above and below the boundary. Since the left-hand expression equals zero, it may be subtracted from the

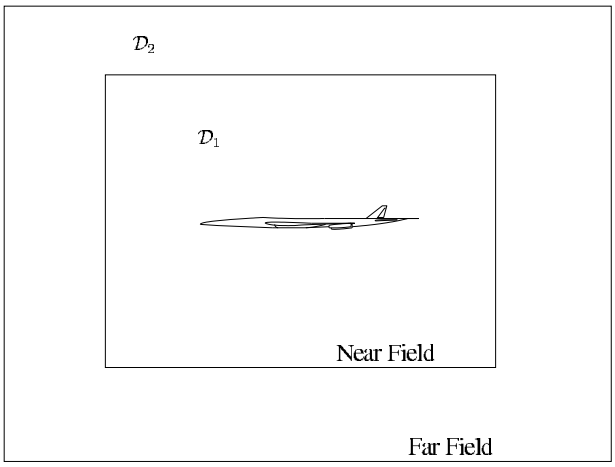


Figure 6.1: Near Field and Far-Field Domains

variation in the cost function (6.2) to give

$$\delta I = \int_{\mathcal{B}_W} \left[\varpi_1 \delta\mathcal{M} - n_i \psi^T \delta F_i\right] d\mathcal{B}_\xi + \int_{\mathcal{B}_{NF}} \left[\varpi_2 \delta\mathcal{N} - n_i \left(\psi^+ - \psi^-\right)^T \delta F_i\right] d\mathcal{B}_\xi + \int_{\mathcal{D}_1} \frac{\partial \psi^T}{\partial \xi_i} \delta F_i d\mathcal{D}_\xi + \int_{\mathcal{D}_2} \frac{\partial \psi^T}{\partial \xi_i} \delta F_i d\mathcal{D}_\xi. \tag{6.6}$$

Since ψ is an arbitrary differentiable function, it may be chosen in such a way that δI no longer depends explicitly on the variation of the state vector δw. The gradient of the cost function can then be evaluated directly from the metric variations without having to re-compute the variation δw resulting from the perturbation of each design variable.

Comparing equations (6.3) and (6.5), the variation δw may be eliminated from (6.6) by equating all field terms with subscript "I" to produce a differential adjoint system governing ψ

$$\left[\frac{\partial \psi^T}{\partial \xi_i} \left[F_{iw}\right]_I\right]_{\mathcal{D}_1+\mathcal{D}_2} = 0 \quad \text{in } \mathcal{D}. \tag{6.7}$$

The corresponding wall and near field adjoint boundary conditions are produced by

equating the subscript "I" boundary terms in equation (6.6) to produce

$$n_i \psi^T \left[F_{iw}\right]_I = \varpi_1 \mathcal{M}_w \quad \text{on } \mathcal{B}_W \tag{6.8}$$

$$n_i \left(\psi^+ - \psi^-\right)^T \left[F_{iw}\right]_I = \varpi_2 \mathcal{N}_w \quad \text{on } \mathcal{B}_{NF}. \tag{6.9}$$

The remaining terms from equation (6.6) then yield a simplified expression for the variation of the cost function which defines the gradient

$$\begin{aligned} \delta I = & \int_{\mathcal{B}_W} \left\{ \varpi_1 \delta \mathcal{M}_{II} - n_i \psi^T \left[\delta F_i\right]_{II} \right\} d\mathcal{B}_\xi \\ & + \int_{\mathcal{B}_{NF}} \left\{ \varpi_2 \delta \mathcal{N}_{II} - n_i \left(\psi^+ - \psi^-\right)^T \left[\delta F_i\right]_{II} \right\} d\mathcal{B}_\xi \\ & + \int_{\mathcal{D}_1 + \mathcal{D}_2} \left\{ \frac{\partial \psi^T}{\partial \xi_i} \left[\delta F_i\right]_{II} \right\} d\mathcal{D}_\xi. \end{aligned} \tag{6.10}$$

The details of the formula for the gradient depend on the way in which the boundary shape is parameterized as a function of the design variables and the way in which the mesh is deformed as the boundary is modified. The boundary conditions satisfied by the flow equations restrict the form of the left-hand side of the adjoint boundary conditions (6.8) and (6.9). Consequently, the boundary contribution to the cost functions $\mathcal{M}$ and $\mathcal{N}$ cannot be specified arbitrarily. Instead it must be chosen from the class of functions which allow cancellation of all terms containing δw in the boundary integral of equation (6.6). In this research the cost function is the weighted sum of the drag coefficient and the Sobolev norm of the difference between the current and target remote pressure distributions. From equation (6.1), $\mathcal{M}$ and $\mathcal{N}$ can be defined as

$$\mathcal{M}(w,S) = \frac{1}{c} C_p \left(\frac{\partial y}{\partial \xi} \cos\alpha - \frac{\partial x}{\partial \xi} \sin\alpha \right) \quad \text{and} \quad \mathcal{N}(w,S) = \frac{1}{2} \left(p - p_T\right)^2.$$

where,

$$\begin{aligned} c &= \text{Wing Chord} \\ C_p &= \text{Pressure Coefficient} \\ \alpha &= \text{Angle of Attack} \\ p &= \text{Current Pressure} \\ p_T &= \text{Target Pressure.} \end{aligned}$$

The cost function can then be written as

$$I = \varpi_1 \frac{1}{c} \int_{\mathcal{B}_W} C_p \left(\frac{\partial y}{\partial \xi} \cos\alpha - \frac{\partial x}{\partial \xi} \sin\alpha \right) d\mathcal{B}_\xi + \varpi_2 \frac{1}{2} \int_{\mathcal{B}_{NF}} (p - p_T)^2 \, d\mathcal{B}_\xi,$$

and further simplified to

$$I = \varpi_1 C_D + \varpi_2 \frac{1}{2} \int_{\mathcal{B}_{NF}} (p - p_T)^2 \, d\mathcal{B}_\xi. \tag{6.11}$$

The values of the weighting coefficients are selected based on the relative magnitude of the gradients of the drag minimization and the remote inverse cost functions. The remote inverse gradient is typically an order of magnitude smaller than the gradient due to drag minimization. Therefore, the weights are chosen to increase the magnitude of the gradient from the remote inverse cost function. In practice, larger weights are used for the remote inverse gradient, since the primary design objective is to reduce the near field pressure signature. The disadvantage of this approach is that the weights must be chosen at the beginning of the design process and if the user does not have prior knowledge of the magnitude of the gradients, then generally an initial guess is taken. The weights are altered for subsequent runs.

An alternative method for problems with more than one objective function is to develop separate adjoint equations, one for each objective function. Both gradients are then calculated separately, multiplied by weights, and summed. A direction of improvement is then based on with the new gradient. This method has the advantage that the user is better equipped with knowledge regarding the difference in magnitude

between the two gradients. Appropriate weights are chosen to achieve the desired compromise. A disadvantage is the need to calculate a separate adjoint solution for each objective function.

In this work, we prefer to use a composite cost function, since we had apriori knowledge regarding the magnitude of the gradient contribution from the remote inverse and the drag minimization cost functions.

6.1.2 Formulation of the Discrete Adjoint Equation for the Remote Inverse Problem

This sub-section explores the discrete adjoint approach for the remote inverse design problem. Unlike the continuous adjoint case, the derivation of the discrete adjoint equation for the remote inverse problem is very similar to that of the viscous inverse and drag minimization problems in Chapter 4 and the unsteady problem formulated in Chapter 5.

The first step is to define the discrete cost function I as,

$$I = \varpi_1 C_D + \varpi_2 \frac{1}{2} \sum_{NF} \left(p_i - p_T\right)^2 \, \Delta s_i, \tag{6.12}$$

where C_D is total wing drag coefficient, p is the current near field pressure, p_T is the target near field pressure, and ϖ_1 and ϖ_2 are weighting coefficients.

The variation of the cost function, δI, can be augmented by the discrete governing equations appropriately pre-multiplied by the adjoint variable $\psi^T_{i,j,k}$

$$\begin{aligned} \delta I = {} & \varpi_1 \delta C_D + \varpi_2 \sum_{NF} \left(p_i - p_T\right) \delta p_i \Delta s_i \\ & + \sum_{i=2}^{nx} \sum_{j=2}^{ny} \sum_{k=2}^{nz} \psi^T_{i,j,k} \delta \left[R\left(w\right) + D\left(w\right)\right]_{i,j,k}. \end{aligned} \tag{6.13}$$

Here the first term represents the discrete drag minimization cost function. The second term represents the discrete remote inverse design cost function evaluated at a near field location which is approximately L chord length's away from the wing, $R(w)$

is the discrete field equation, and $D(w)$ is the discrete artificial dissipation term. In order to eliminate δw from equation (6.13), terms multiplied by the variation $\delta w_{i,j,k}$ of the discrete flow variables are collected and equated to zero. The procedure to obtain the discrete adjoint is identical to that shown in section 4.3 and will not be repeated here. The discrete adjoint equations from Chapter 4 are extended to three-dimensions to be used in this part of the research. A three-dimensional re-derivation of the convective and artificial dissipative discrete adjoint fluxes would be a difficult assignment. For the convective discrete adjoint flux, a total of seven cells are required to form the flux. In the case of the discrete artificial dissipation flux, thirteen cells are needed. The additional complexity is added by the fact that each cell has six faces. Instead of a re-derivation of the discrete adjoint fluxes, the three-dimensional fluxes are obtained by a careful study of the two-dimensional discrete adjoint fluxes for the convective and dissipative components. Since obvious patterns exists within the two-dimensional framework, an extension to three-dimensions is easily accomplished. A three-dimensional version of the discrete viscous adjoint flux will not be formulated since the remote inverse problem is strictly limited to an inviscid case. If it is desired, an extension to three-dimensions will be similar to the procedures used to obtain the three-dimensional discrete convective and artificial dissipation adjoint fluxes.

Discrete Adjoint Boundary Condition

We now discuss the discrete adjoint boundary condition for the calculation of remote sensitivities for supersonic flow. The $\delta w_{i,NF}$ term from the discrete cost function is added to the corresponding term from equation (4.27). The discrete boundary condition appears as a source term in the adjoint fluxes. For example, at cell (i, NF) the adjoint equation in two dimensions can be discretized as follows,

$$
\begin{aligned}
V\frac{\partial \psi_{i,NF}}{\partial t} = &-\frac{1}{2}A^T_{i-\frac{1}{2},NF}\left(\psi_{i,NF}-\psi_{i-1,NF}\right)\\
&-\frac{1}{2}A^T_{i+\frac{1}{2},NF}\left(\psi_{i+1,NF}-\psi_{i,NF}\right)\\
&-\frac{1}{2}B^T_{i,NF+\frac{1}{2}}\left(\psi_{i,NF+1}-\psi_{i,NF}\right)\\
&-\frac{1}{2}B^T_{i,NF-\frac{1}{2}}\left(\psi_{i,NF}-\psi_{i,NF-1}\right)+\Phi_{NF},
\end{aligned}
\tag{6.14}
$$

where V is the cell area, Φ_{NF} is the source term for inverse design,

$$\Phi_{NF} = -\varpi_2(p-p_T)\Delta s_i \delta p_{i,NF},$$

and

$$A^T_{i+\frac{1}{2},NF} = \Delta y_{\eta_{i+\frac{1}{2},NF}}\left[\frac{\partial f}{\partial w}\right]^T_{i,NF} - \Delta x_{\eta_{i+\frac{1}{2},NF}}\left[\frac{\partial g}{\partial w}\right]^T_{i,NF}.$$

The boundary condition for the drag minimization cost function is identical to that derived in Chapter 4 and will not be shown here. The only addition is the multiplication of the drag source term by the weighting coefficient ϖ_1. Application of the coupled drag minimization and remote inverse problem will in effect introduce two source terms into the discrete adjoint fluxes. The first is added to the interior points to satisfy the remote inverse problem and the second is added to the cells above the surface to satisfy the drag minimization problem.

6.2 Implementation of Remote Inverse Design

The remote inverse design procedure follows the outline presented in section 4.9. The following algorithm describes the procedure to locate and calculate the discrete adjoint boundary condition source terms.

Phase 1

for $k = 2, \cdots, k_{maxcut}$
  calculate cell centers
  calculate target pressure.
    for $i = 1, \cdots,$ numpoints
      Calculate pressure at target location using a bilinear interpolation.
      Calculate the difference between the current and target pressure.
    end i−loop
  find near field cell
end k−loop

Phase 2

for $k = 1, \cdots, k_{maxcut} - 1$
  for $i = 1, \cdots,$ numpbnd
    Calculate remote inverse source term using trilinear interpolation.
  end i−loop
end k−loop

The search-locate-calculate remote inverse source term algorithm is completed in two phases. In the first phase, the goal is to calculate the pressure at the target location and search for the closest cell to the specified target location at each wing span location. This cell is defined as the near field cell. The first step at phase one is to calculate the cell centers of all cells in the current span location. These are needed by the bilinear interpolation routine to produce weighting coefficients to calculate the pressure at the target location. Once the current pressure is known at the target

location, the first variation of the remote inverse cost function is calculated which amounts to taking the difference between the current and target pressure values. This step is repeated *numpoints* times, where *numpoints* is the total number of target locations. The next important step in this procedure is to search for the closest cell to the target location. This is achieved by calculating the location of the diagonal of each cell that surrounds the target point and then performing a search and compare routine to locate the near field cell. The entire process is repeated for each span station from $k = 2, \ldots, k_{maxcut}$, where k_{maxcut} is the tip span station.

The second phase of the search-locate-calculate algorithm is to compute the value of the source term at the newly found near field cell. Since the variation of the remote inverse cost function is calculated at the target location in phase one, a trilinear interpolation routine is needed to transfer the value to the near field cell. The bilinear and trilinear interpolation routines are provided by Saunders [75].

The search-locate-calculate algorithm is repeated at every design cycle, since the possibility exists that the mesh may have been modified and thus a new location for the source term may be required. Figure 6.2 shows the location of the near field target pressure (+) and the adjoint remote sensitivity source terms (o).

6.3 Results

This section presents the results of an inverse Ni-bump design problem, and inverse biconvex airfoil, inverse biconvex wing, and coupled remote inverse and drag minimization for three-dimensional wings in supersonic flow. The objective here is to demonstrate the capability of the remote adjoint problem for the various geometries and design parameters. The calculations are performed with a modified version of Jameson's SYN88 software, which augments the FLO88 flow solver with an adjoint solver, and shape modification procedure.

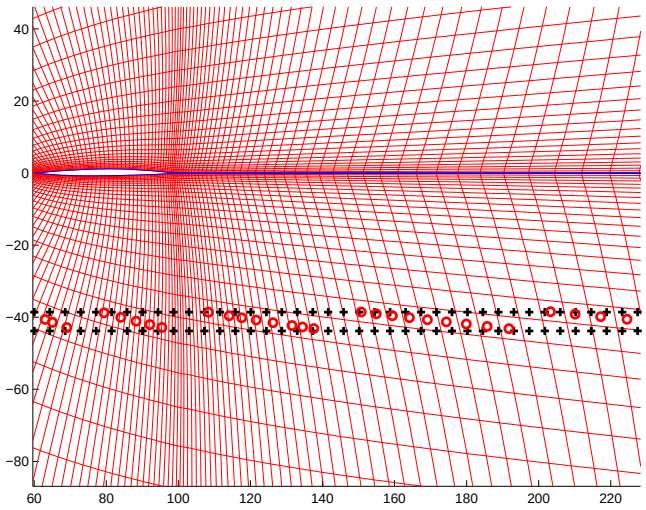


Figure 6.2: Location of Near Field Pressure and Adjoint Remote Sensitivity Source Terms

6.3.1 Ni-Bump

As a first test of the new method, we constructed the following test problem, based on the Ni-bump geometry with a longer downstream portion of the channel. A channel of unit height and length, $l = 8.0$ is constructed. A 1.8% thick Ni-bump of unit chord is centered about $x = 6.0$. Along the upper wall, a target pressure corresponding to the presence of the same Ni-bump centered about $x = 2.0$ is specified, and the geometry of the complete lower surface of the channel is allowed to move so that the target pressure is obtained. Clearly, the solution of this problem is the disappearance of the initial bump, and the formation of the exact same bump centered about $x = 2.0$.

Figure 6.3 illustrates the non-dimensional pressure contours of a typical Euler solution on the final configuration. Figure 6.4 shows the initial and final (red and blue respectively) geometries of the lower wall as explained above. Notice that the aspect ratio of the figure on the right has been modified so that the small thickness Ni-bumps are visible. Notice the complete disappearance of the original bump and the formation of the new bump. Figures 6.5 and 6.6 illustrate the initial and final

lower surface pressure distributions. The blue line in figure 6.5 is labeled "target" but this is not to be confused with the target pressure distribution that is introduced into the adjoint boundary condition to control the shape of the bump. The word target here signifies the desired pressure distribution if the desired Ni-bump is created to the expected geometry and location along the channel. Figure 6.6 illustrates the creation of the new bump and its lower pressure distribution. Apart from a few discrepancies at $x = 3.0$ and $x = 5.0$, there is a perfect match between the initial and final lower surface pressure distributions.

A more remarkable set of plots are those of the pressure distributions on the upper surface of the channel before and after the optimization process has been completed. Figure 6.7 illustrates the initial pressure distribution corresponding to the Ni-bump centered about $x = 6.0$ (in red), while the target pressure distribution along the upper surface of the bump is in blue. After 60 design iterations, using the adjoint procedure described above for the computation of the sensitivities, Figure 6.8 illustrates that the target pressure distribution along the upper surface is very closely matched. The Ni-bump centered about $x = 6.0$ has completely disappeared, while its twin centered about $x = 2.0$ has formed.

These results verify the feasibility of the remote adjoint method. The next step is to apply this approach to the sonic boom minimization problem by modifying the near field pressure distribution in such a way that it achieves a pre-specified target.

6.3.2 Biconvex Wing: Verification Study

The next test is for three-dimensional flow. We specify a known near field pressure distribution that is realizable as the target pressure by taking the result of a direct calculation for a given shape. Then we initialize the geometry with another shape, and try to recover the original shape from the target pressure distribution.

The test problem has been constructed based on a biconvex wing with a 3% thickness ratio at the root and 1.5% at the tip. The leading edge sweep of the wing is 7.125 degrees. The aspect ratio is 3.0 with a 0.218 taper ratio. To begin the remote inverse design process, the near field pressure distribution for a biconvex wing

with a 2% thickness ratio at the root and 1% at the tip is first calculated in order to provide a realizable target pressure. All other wing geometry parameters remain unchanged. The flow solution is obtained at a Mach Number of 1.5, an angle of attack of 0 degrees, and the near field target pressure is computed at a distance of one chord length below the surface of the wing. We then perform an inverse calculation with this target, starting from the 3% biconvex wing. Clearly, the solution of this problem is the reproduction of the 2% biconvex wing that initially produced the target pressure distribution.

Figures 6.9(a-b) illustrate the target (+), initial ($\Box$), and final ($*$) near field pressure distributions at the root and mid-span sections of the wing after 50 design cycles. An almost perfect point-to-point match is achieved. The objective function is the integral of the square of the difference between the current and target near field pressure. No lift or thickness constrains are enforced. These results clearly validate the remote adjoint method for three-dimensional supersonic flow.

6.3.3 Biconvex Wing: Near Field Pressure Reduction, Without Constraints

In order to illustrate the possibility of near field pressure reduction, a target pressure distribution is obtained by re-scaling the initial near field pressure distribution. Although this near field target pressure may not be realizable, the hope is that a reduced near field signature may result in lower ground boom signature. Ultimately, this step will be replaced by a method that produces a target near field pressure based upon the desired ground pressure signature.

The target pressure is obtained using the FLO88 flow solver on a biconvex wing with a 5% thickness ratio at the root and 3.25% at the tip at a flight condition of Mach 1.5 and a lift coefficient of $C_L = 0.1$ on a 192x32x32 C-grid. All other wing geometry parameters are the same as the previous section. The target pressure is reduced by 40% of its original value.

Figures 6.10(a-b) illustrate the target (+), initial ($\Box$), and final ($*$) near field pressure distribution. After 50 design cycles the final near field pressure almost

matches the target near field pressure. Neither the lift nor the thickness ratio at each span station are constrained.

6.3.4 Biconvex Wing: Near Field Pressure Reduction, With Constraints

We now repeat the same design case but with the lift and thickness constrained. The value of the lift coefficient is maintained by adjusting the angle of attack to attain the desired lift coefficient. The thickness ratio at each span station is forced to remain the same.

The resulting solution is very different from the previous case. Figures 6.11(a-b) illustrate the baseline and optimized airfoil sections at span stations $z = 0.0957$ and $z = 0.571$. The unexpected result in this test problem is the shape of the lower surface of the wing. At both the root and mid-span sections, the lower leading edge slope has slightly increased but contains a larger expansion region. It is this modification that allows the near field wing peak pressure to be reduced. The slightly larger slope at the leading edge increases the strength of the attached shock. But it is the larger expansion region that weakens the strength of the attached shock in the near field region.

Figures 6.12(a-b) show the target, initial, and final near field pressure distributions. The desired target pressure distribution is not achieved in contrast with the unconstrained case illustrated in Figure 6.10. In this case, there is a struggle between the near field peak pressure reduction and the achievement of constant lift. Each design cycle, produces a shape modification that shifts the near field pressure distribution towards the target pressure. Unfortunately, this also causes a reduction in the lift coefficient. This must be compensated by an increase in the angle of attack to maintain the total lift coefficient, which in turn leads to an increase in the near field peak pressure. After 50 design cycles, the solution converges to the $(- * -)$ line in Figure 6.12. The final peak pressure has been reduced to almost 23% its original value at the root section and 18% at the mid-span section.

6.3.5 Highly Swept Blunt LE Wing: Near Field Pressure and Drag Reduction

Finally, we examine a highly swept wing with a blunt subsonic leading edge. Blunt leading edge symmetric airfoils are created with a thickness ratio of 4% at each span station. The wing has 60 degrees of leading edge sweep, and a 0.56 taper ratio.

Figures 6.13(a-b) illustrate the airfoil sections at span locations $z = 0.0957$ and $z = 0.571$. Each figure contains three plots: Baseline Wing, Wing Optimized with Remote Inverse Objective Function, and Wing Optimized with Remote Inverse and Drag Minimization Objective Functions. The airfoil sections are scaled to exaggerate the modifications. Figures 6.14(a-b) show the corresponding near field pressure distribution at the same span location. At the root section, a 14.2% reduction in the peak pressure is obtained with the remote inverse as the objective function and a slightly smaller reduction of 13.0% is achieved with the weighted sum of the remote inverse and drag minimization cost functions. At the mid span location, a 10.4% reduction is achieved with the former and 9.4% reduction with the latter cost function. For the case of joint drag minimization and remote inverse design, the weights are $\varpi_1 = 0.005$ and $\varpi_2 = 1.0$. The weighting coefficient for the drag minimization cost function is smaller due to the larger magnitude of its gradient contribution. Since the main objective of the design is to reduce the near field peak pressure, the weights were adjusted to achieve the desired result. A comparison of the two final designs achieved by the remote inverse and the simultaneous remote inverse and drag minimization cost functions show that the two methods provide similar results, with the remote inverse cost function providing a slightly greater reduction in the near field pressure peak.

Design Case	C_D	$C_D(\%)$
Baseline	0.0038	
Remote Inverse	0.0040	+3.4%
Drag Minimization	0.0032	−16.2%
Remote Inverse and Drag Min.	0.0036	−5.8%

Table 6.1: Total Wing Drag Coefficient for Various Design Cases

Table 6.1 lists the total wing drag coefficient for the baseline wing and the optimized wings. If only the remote inverse cost function is used, then a 3.4% increase in total drag is seen versus the 5.8% reduction in total drag with the dual objective function case. The reason for the difference between the two final designs can be seen clearly in Figures 6.15 and 6.16. Figure 6.15(a) illustrates the pressure contours on the upper surface of the wing. A strong shock is present slightly aft of the mid chord location at the wing tip. The shock strength weakens as it moves inboard. Figure 6.15(b) shows the pressure contours of the final design using only the remote inverse cost function. Here we notice the presence of the shock wave at the wing tip. In Figure 6.16(c) this shock is eliminated in the design case using only the drag minimization objective function. However, in Figure 6.16(d) it is clearly seen that the wing tip shock strength is reduced thus contributing to the reduction in the total drag coefficient.

Figures 6.17(a-c) illustrate the coefficient of pressure plots for the baseline and the two redesigned wings at three different span locations. At $z = 0.85$, the plots clearly show the reduction in the strength of the shock.

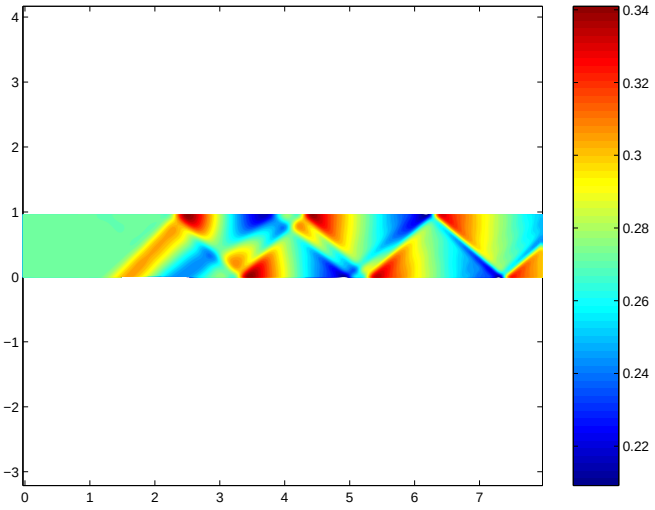


Figure 6.3: Final Solution Pressure Contours for Ni-Bump Geometry

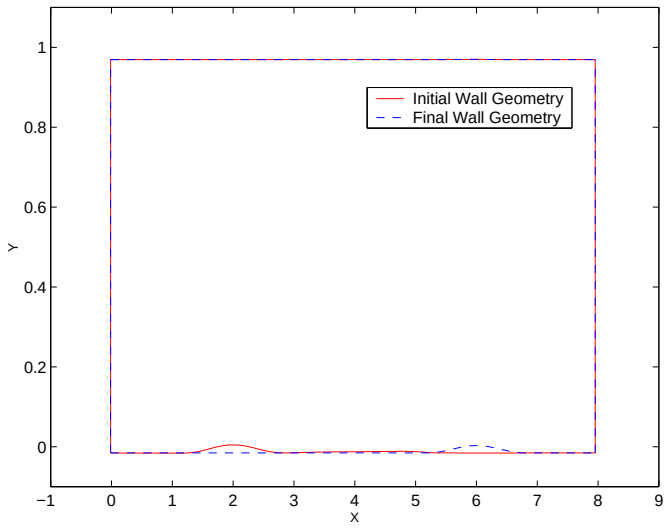


Figure 6.4: Initial and Final Lower Ni-bump Wall Geometry

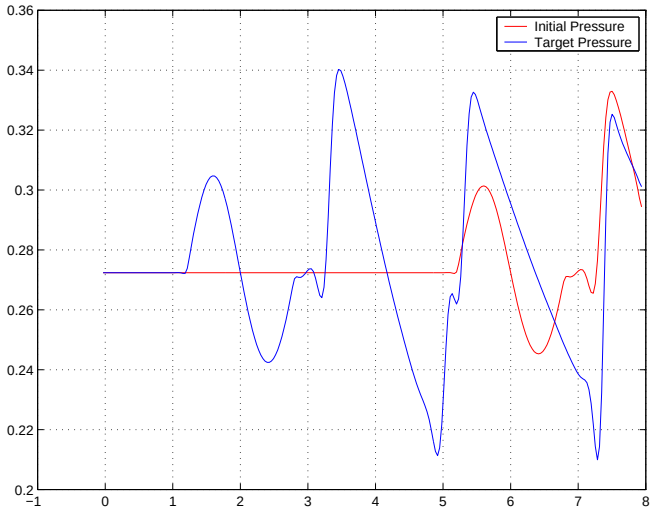


Figure 6.5: Initial Lower Surface Pressure Distributions.

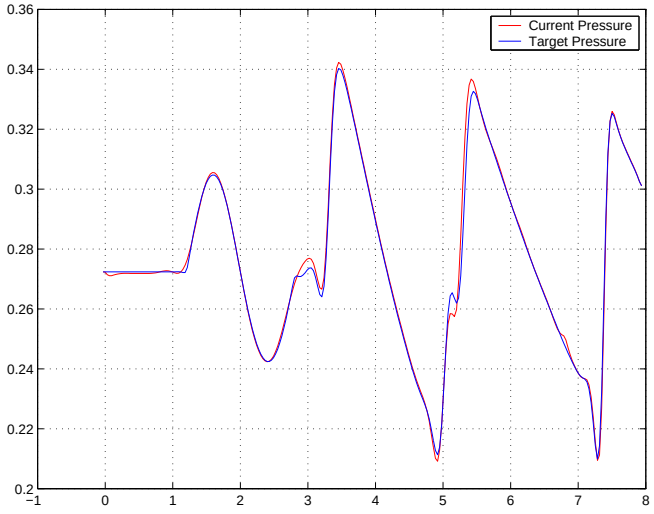


Figure 6.6: Final Lower Surface Pressure Distributions.

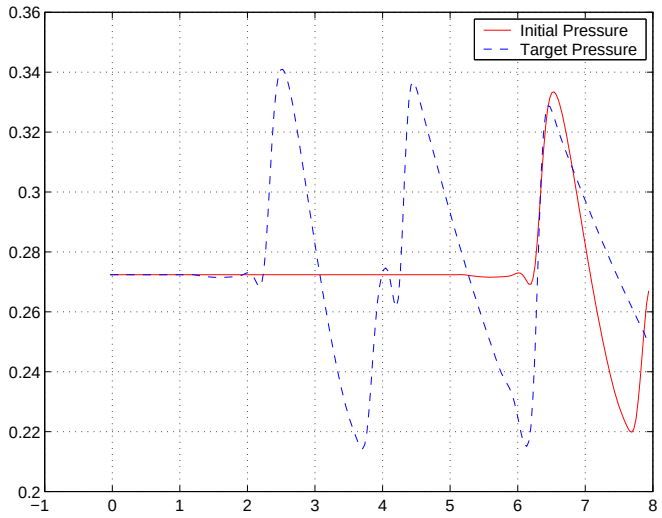


Figure 6.7: Initial Upper Surface Pressure Distributions.

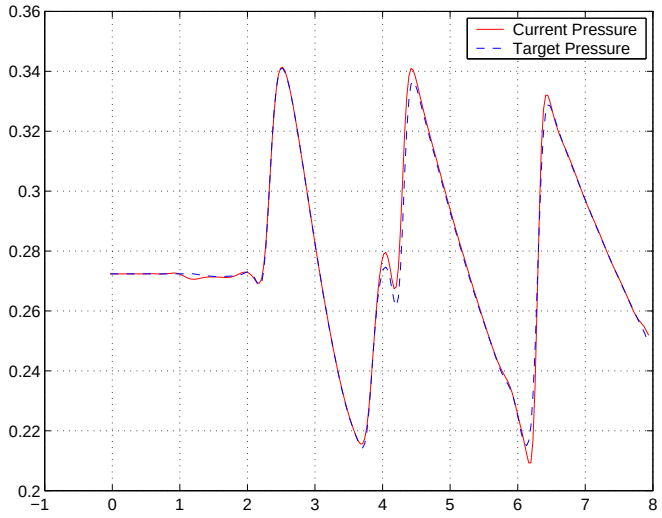
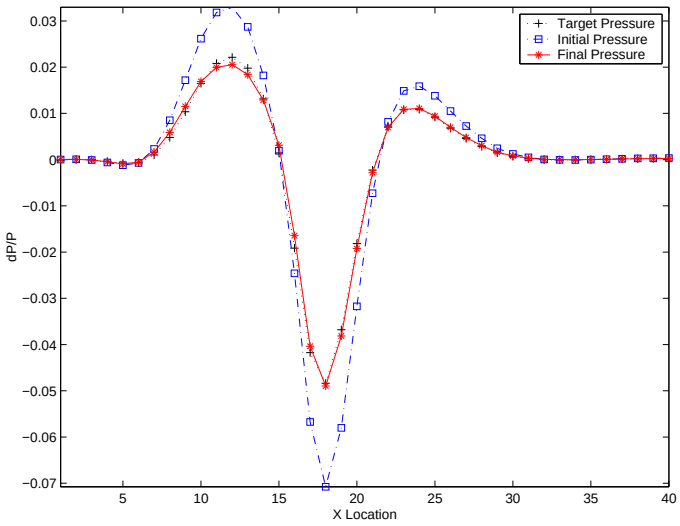
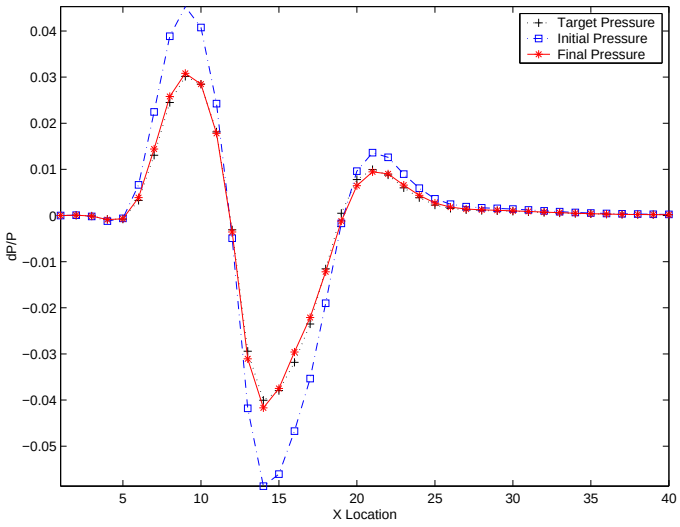


Figure 6.8: Final Upper Surface Pressure Distributions.

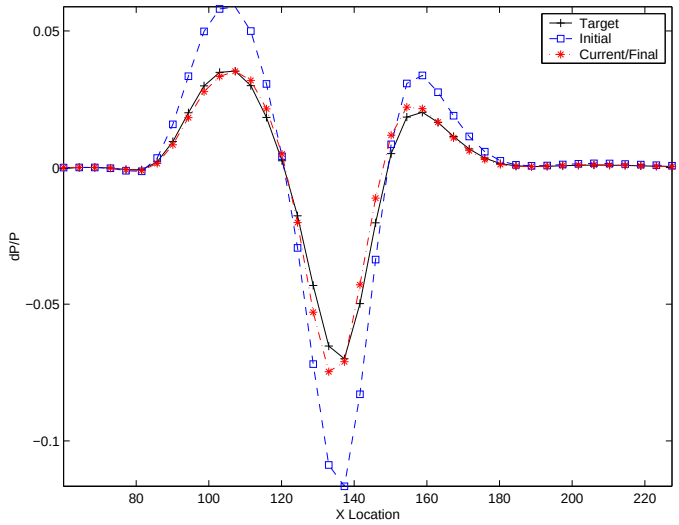


9a: z = 0.0957

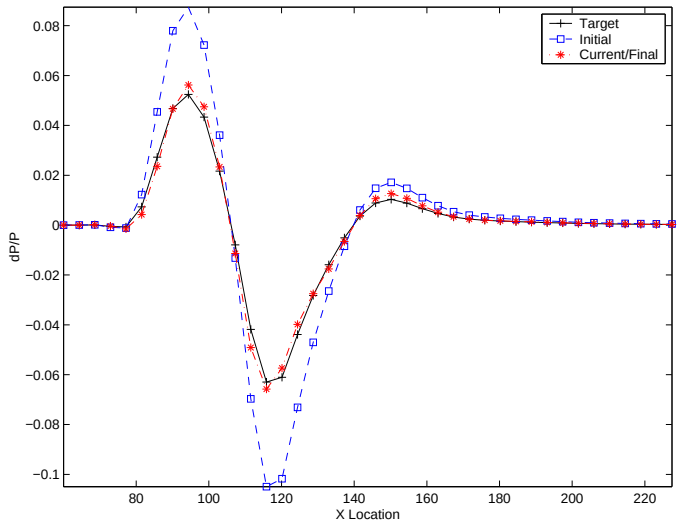


9b: z = 0.571

Figure 6.9: Verification Study: Target, Initial, and Final Near Field Pressure Distribution. Mach = 1.5, $\alpha = 0$ deg.

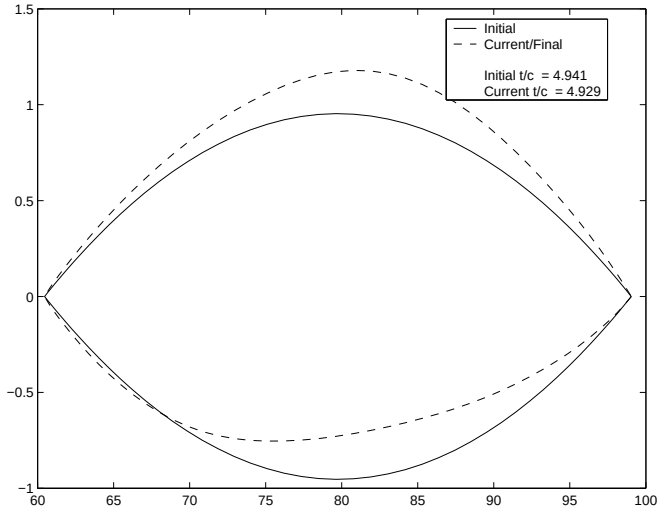


10a: z = 0.0957

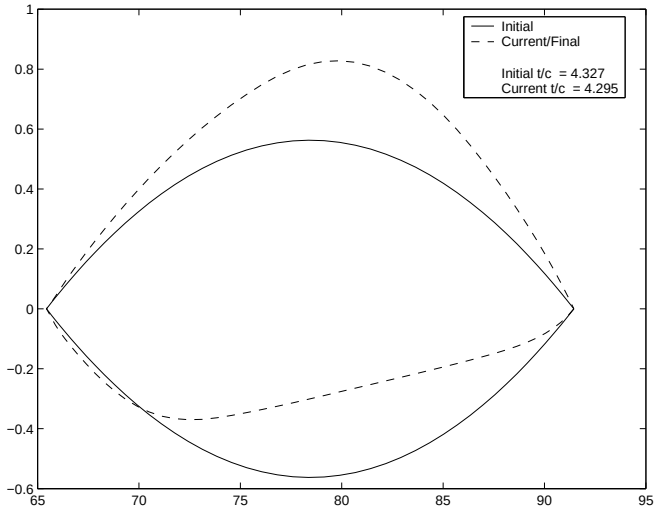


10b: z = 0.571

Figure 6.10: Sonic Boom Minimization: Target, Initial, and Final Near Field Pressure Distribution after 50 Design Cycles. Mach = 1.5, $\alpha = 1.75$ deg., No Lift Coefficient and Thickness Ratio Constraints

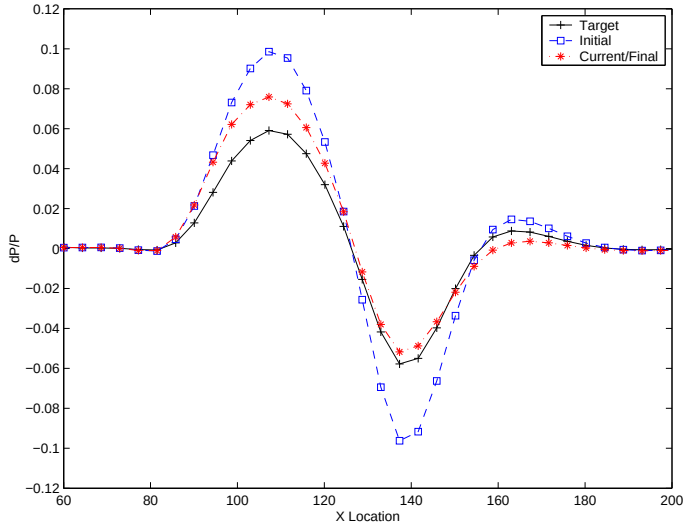


11a: z = 0.0957

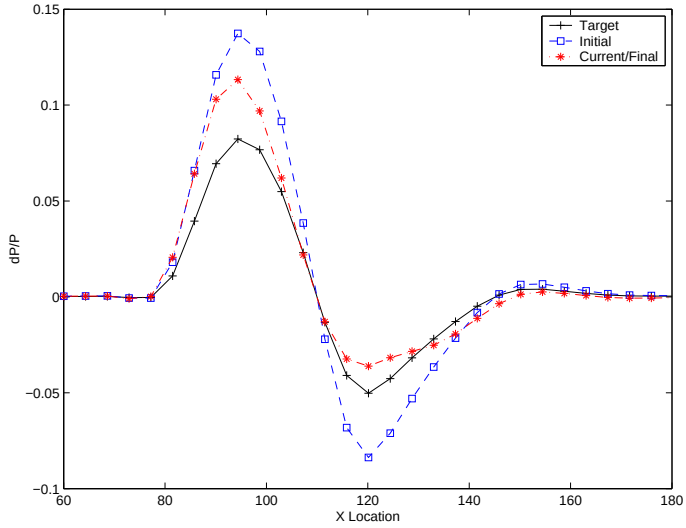


11b: z = 0.571

Figure 6.11: Sonic Boom Minimization: Initial and Final Airfoil Shape After 50 Design Cycles. Mach = 1.5, $\alpha = 1.75$ deg, Fixed Lift Coefficient = 0.1, Fixed Thickness Ratio

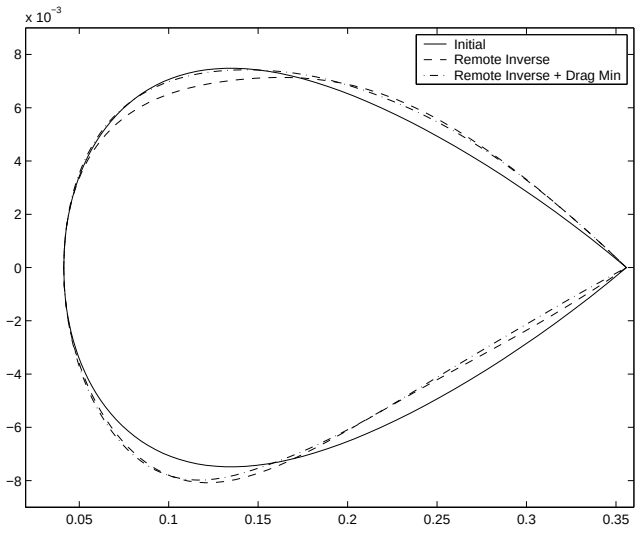


12a: z = 0.0957

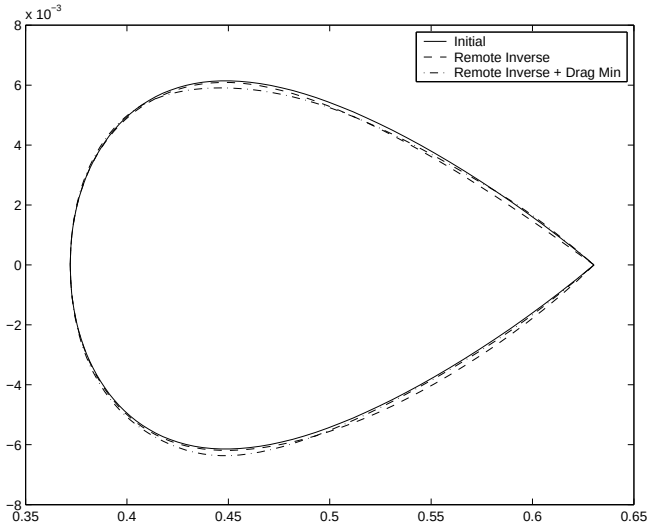


12b: z = 0.571

Figure 6.12: Sonic Boom Minimization: Target, Initial, and Final Near Field Pressure Distribution after 50 Design Cycles. Mach = 1.5, α = 1.75 deg., Fixed Lift Coefficient = 0.1, Fixed Thickness Ratio

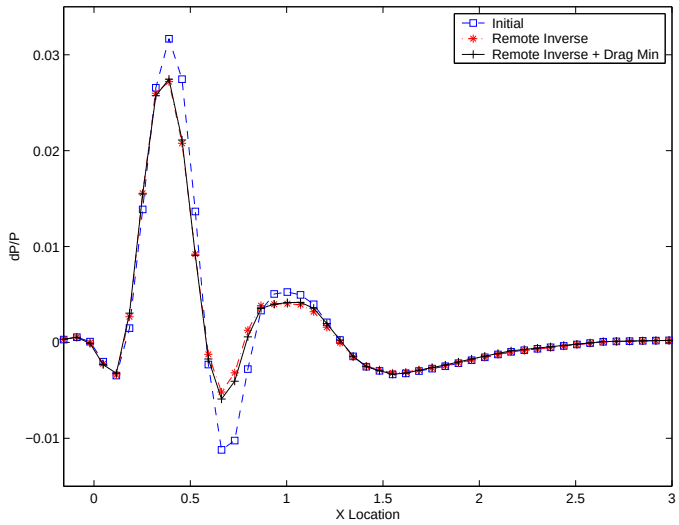


13a: z = 0.0957

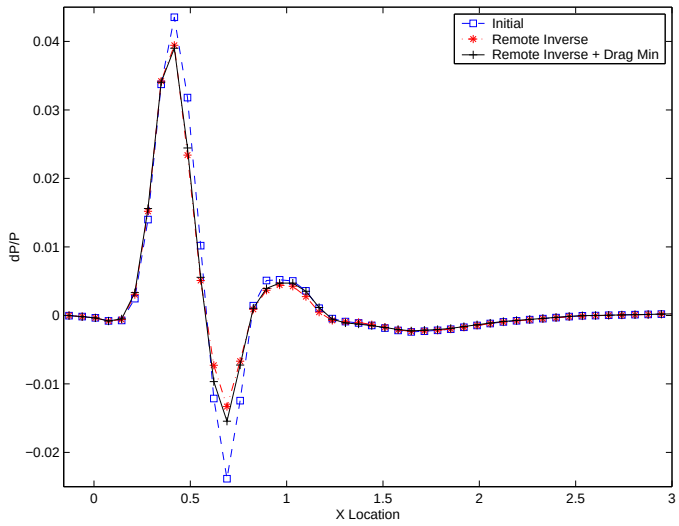


13b: z = 0.571

Figure 6.13: Sonic Boom and Drag Minimization: Initial and Final Airfoil Shape After 50 Design Cycles. Mach = 1.5, α = 0.829 deg, Fixed Lift Coefficient = 0.05, Fixed Thickness Ratio

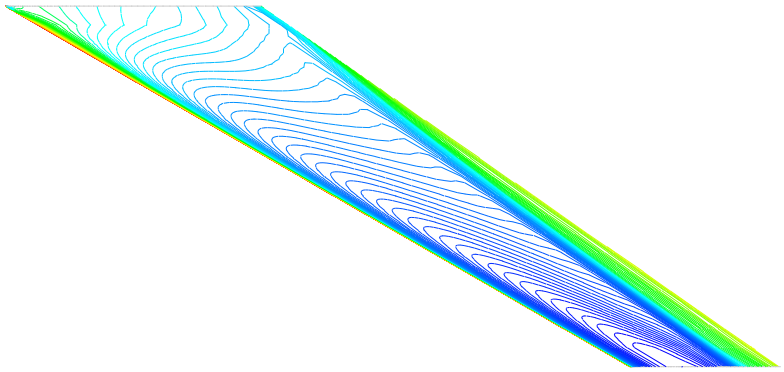


14a: z = 0.0957

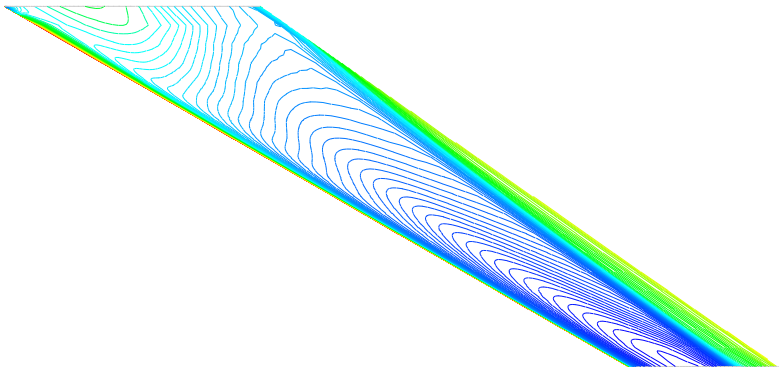


14b: z = 0.571

Figure 6.14: Sonic Boom and Drag Minimization: Target, Initial, and Final Near Field Pressure Distribution after 50 Design Cycles. Mach = 1.5, α = 0.829 deg., Fixed Lift Coefficient = 0.05, Fixed Thickness Ratio

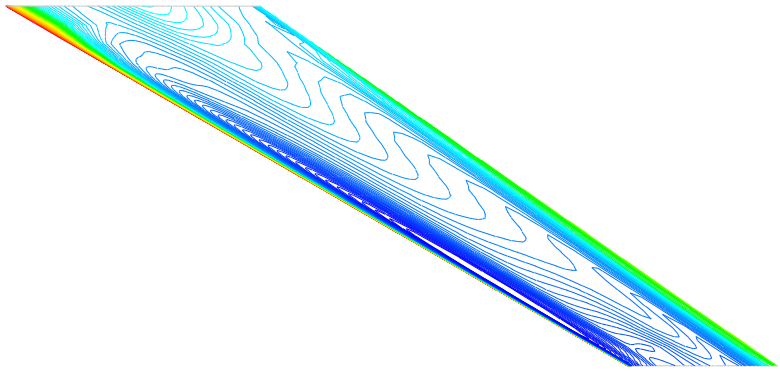


15a: Base Wing

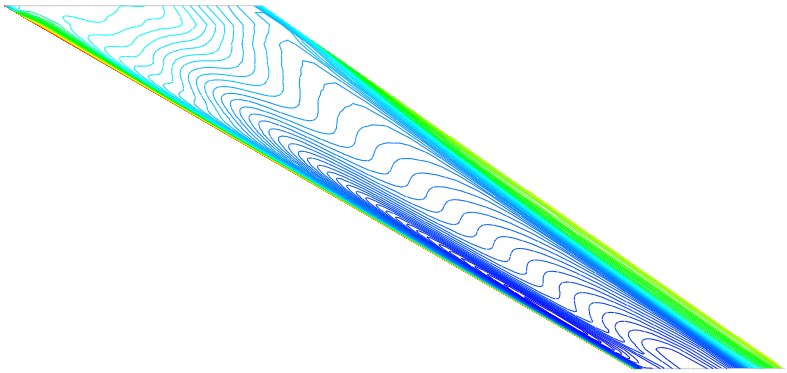


15b: Optimized with Remote Inverse Design Objective Function

Figure 6.15: Sonic Boom and Drag Minimization: Pressure Contours of Final Design of the Upper Surface of the Wing. Mach = 1.5, $\alpha = 0.829$ deg, Fixed Lift Coefficient = 0.05, Fixed Thickness Ratio

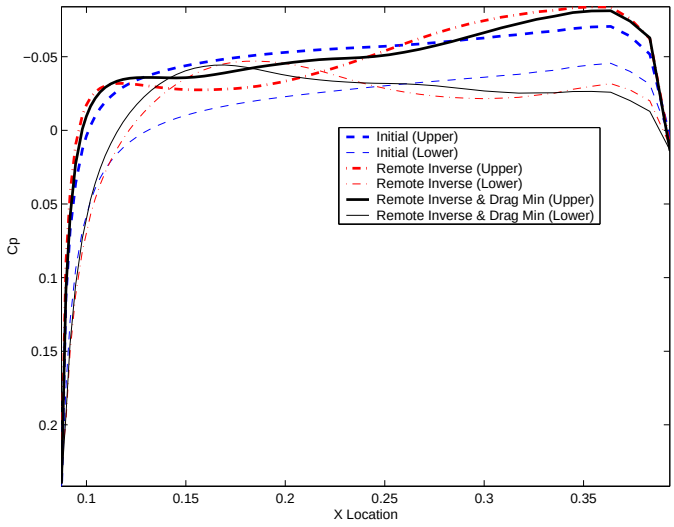


16a: Optimized with Drag
Minimization Objective Function

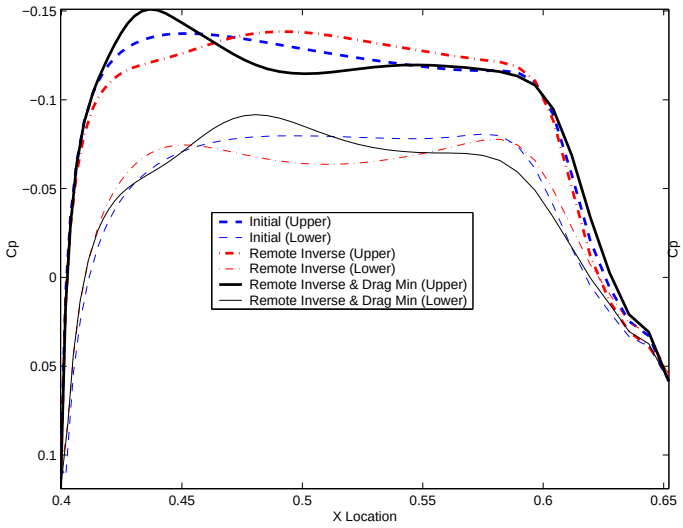


16b: Optimized with Remote Inverse
Design and Drag Minimization
Objective Functions

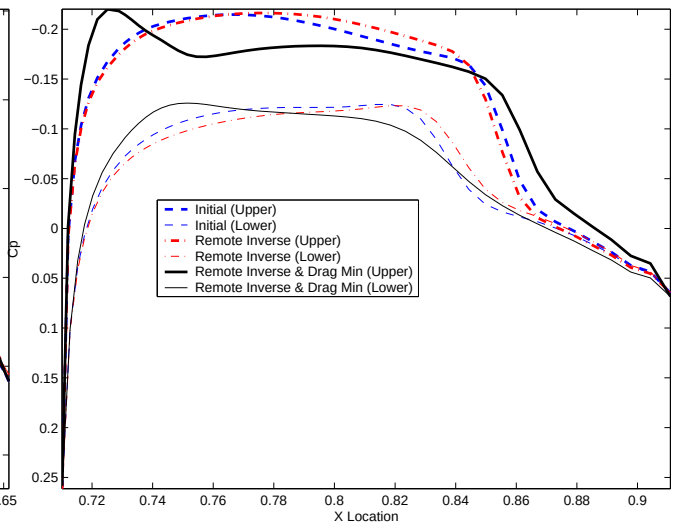
Figure 6.16: Sonic Boom and Drag Minimization: Pressure Contours of Final Design of the Upper Surface of the Wing. Mach = 1.5, $\alpha = 0.829$ deg, Fixed Lift Coefficient = 0.05, Fixed Thickness Ratio



17a: z = 0.0957



17b: z = 0.5



17c: z = 0.85

Figure 6.17: Sonic Boom and Drag Minimization: Surface Pressure Coefficient. Mach = 1.5, $\alpha = 0.829$ deg, Fixed Lift Coefficient = 0.05, Fixed Thickness Ratio

Chapter 7

Conclusions

This chapter presents the main conclusions which may be drawn from the three focal areas of research in this work: first, the development of the discrete viscous adjoint equation; second, the formulation of the time accurate continuous and discrete adjoint equations; third, the implementation of the adjoint approach for the calculation of remote sensitivities.

7.1 Discrete Versus Continuous Adjoint Approaches

The following are the conclusions of the comparison between the discrete and continuous adjoint approaches.

In the limit as the mesh width is reduced to zero, the discrete adjoint convective and viscous fluxes both reduce to the continuous adjoint fluxes. The major difference between the two approaches is the manner in which the adjoint boundary condition is introduced into the system of equations. The continuous adjoint boundary condition appears as an update to the costate values below the wall for a cell-centered scheme, and the discrete adjoint boundary condition appears as a source term in the cell above the wall. As the mesh width is reduced, one recovers the continuous adjoint boundary condition from the discrete adjoint boundary condition for the inviscid and viscous inverse design problems.

A comparison of the gradients from each approach to the finite-difference and

complex-step gradients leads to the following conclusions. First, the discrete adjoint gradients have better agreement than continuous adjoint gradients with complex-step gradients as expected, but the difference is generally small. Figures 4.36-4.38 illustrate this point. Second, as the mesh interval is reduced, both the continuous adjoint gradients and the discrete adjoint gradients approach the complex-step gradients. Third, the difference between the continuous and discrete gradients decreases as the mesh interval is reduced. Table 4.3 verifies this outcome.

As stated in Chapter 4 only a partial discretization of the artificial dissipation scheme was used to produce the discrete adjoint artificial dissipation flux. The results show that the difference is negligible and does not affect the convergence of the optimization algorithm.

Another remarkable property of the control theory approach to optimization is the insensitivity of the accuracy of the adjoint gradient to the convergence of the flow and adjoint solutions. This is an added benefit over the traditional finite-difference approach, which not only requires the re-computation of the flow solution for each design variable, but also requires at least three to four orders of convergence in the flow solutions. In the viscous case, a greater degree of convergence of the flow and adjoint solutions is needed to obtain accurate gradient information. This is due mainly to the fact that the boundary layer requires a greater amount of time to develop.

The cost of deriving the discrete adjoint is greater, but it may provide a route to improving the boundary conditions for the continuous adjoint for viscous flows. The best compromise may be to use the continuous adjoint formulations in the interior of the domain and the discrete adjoint boundary condition.

7.2 Optimum Shape Design for Unsteady Flows

The results presented in Chapter 5 confirm that optimum shapes for periodic unsteady flows can be calculated by the adjoint method with quite moderate computational costs. The time-averaged drag of the RAE 2822 was reduced by 57% at a reduced frequency of 0.450 while maintaining the time-averaged lift coefficient. This benefit was obtained by a drastic reduction of the upper surface curvature, an illustration of

the capability of the method not only to improve the design, but also to provide insight into the kind of modifications needed to improve the performance. Optimization of the airfoil shape for various reduced frequencies and freestream Mach numbers demonstrated the robustness of the adjoint approach. Similar results were obtained for the VR-7 rotor airfoil. However, the improvements were not as large, perhaps because this is already an airfoil that is well suited to rotorcraft.

The results also show that there are large benefits to modeling the unsteady flow. A comparison between the unsteady solutions of airfoils designed using the full unsteady optimization approach and the multipoint design approach showed a 9% benefit of the full unsteady over the multipoint technique at a reduced frequency of 0.450. The advantage decreases at lower reduced frequencies. A comparison of the gradients produced by the time-averaged-flow steady-adjoint, partial unsteady, and full unsteady adjoint methods indicates that the gradients are not sensitive to the method by which the adjoint equations are modeled for the pitching airfoil problem. In fact modeling the unsteady flow and only using the time-averaged flow solution is adequate for this problem.

7.3 Remote Inverse Design Problem

The results demonstrate the feasibility of remote inverse calculations using the adjoint method, which would be impossible with other inverse methods such as CDISC[11]. When combined with a method to identify appropriate near field pressure distributions, the remote adjoint method could be a valuable tool towards the realization of efficient supersonic transports with reduced sonic boom signatures.

7.4 Future Work

The body of work presented in this thesis has demonstrated the successful application of the control theory approach to aerodynamic shape optimization. The formulation of the time dependent optimal design problem and the feasibility of reducing the near field pressure signature through the remote inverse adjoint approach are very much

in their preliminary stages. Some directions for future research are listed below.

1. Extension of the time dependent optimal design problem to the design of multi-stage turbomachinery blades. A substantial portion of the study would need to be focused on developing cost functions and constraints to reduce losses incurred in compressors, improve the aeroelastic stability, and the acoustic response of the turbomachine.

2. Implementation of the remote inverse adjoint method for full aircraft configurations. The intention would be to develop a capability to design efficient low sonic boom supersonic transports. The lift-to-drag ratio, range, and other important performance measures would be included in the formulation of the design problem.

3. Investigation of new convergence acceleration methods for viscous dominated flows. Fast solvers currently available for inviscid flows approach text-book multigrid convergence. An extension of these methods to viscous flows would facilitate the successful design of turbomachinery and helicopter blades.

4. Implementation of the adjoint equation for the nonlinear frequency domain technique. Current research has shown the promise of the frequency domain method for periodic flows.

These projects could all reinforce the role of computational fluid dynamics in the design of future aircraft and turbomachinery.

Appendix A

Viscous Continuous Adjoint Equations

This section illustrates application of control theory to aerodynamic design problems for the case of two-dimensional airfoil design using the Navier-Stokes equations as the mathematical model.

In computational coordinates, the viscous terms in the Navier–Stokes equations have the form

$$\frac{\partial F_{vi}}{\partial \xi_i} = \frac{\partial}{\partial \xi_i} \left(S_{ij} f_{vj} \right) .$$

Computing the variation δw resulting from a shape modification of the boundary, introducing a Lagrange vector ψ and integrating by parts following the steps outlined by equations (4.3) to (6.5) produces

$$\int_{\mathcal{B}} \psi^T \left(\delta S_{2j} f_{vj} + S_{2j} \delta f_{vj} \right) d\mathcal{B}_\xi - \int_{\mathcal{D}} \frac{\partial \psi^T}{\partial \xi_i} \left(\delta S_{ij} f_{vj} + S_{ij} \delta f_{vj} \right) d\mathcal{D}_\xi,$$

where the shape modification is restricted to the coordinate surface $\xi_2 = 0$ so that $n_1 = 0$, and $n_2 = 1$. Furthermore, it is assumed that the boundary contributions at the far field may either be neglected or else eliminated by a proper choice of boundary conditions as previously shown for the inviscid case [35].

The viscous terms will be derived under the assumption that the viscosity and

heat conduction coefficients μ and k are essentially independent of the flow, and that their variations may be neglected. This simplification has been successfully used for many aerodynamic problems of interest. In the case of some turbulent flows, the possibility exists that the flow variations could result in significant changes in the turbulent viscosity, and it may then be necessary to account for its variation in the calculation.

A.1 Transformation to Primitive Variables

The derivation of the viscous adjoint terms is simplified by transforming to the primitive variables

$$\tilde{w}^T = (\rho, u_1, u_2, p)^T,$$

because the viscous stresses depend on the velocity derivatives $\frac{\partial u_i}{\partial x_j}$, while the heat flux can be expressed as

$$\kappa \frac{\partial}{\partial x_i}\left(\frac{p}{\rho}\right)$$

where $\kappa = \frac{k}{R} = \frac{\gamma\mu}{Pr(\gamma-1)}$. The relationship between the conservative and primitive variations is defined by the expressions

$$\delta w = M\delta\tilde{w}, \quad \delta\tilde{w} = M^{-1}\delta w$$

which make use of the transformation matrices $M = \frac{\partial w}{\partial \tilde{w}}$ and $M^{-1} = \frac{\partial \tilde{w}}{\partial w}$. These matrices are provided in transposed form for future convenience

$$M^T = \begin{bmatrix} 1 & u_1 & u_2 & \frac{u_i u_i}{2} \\ 0 & \rho & 0 & \rho u_1 \\ 0 & 0 & \rho & \rho u_2 \\ 0 & 0 & 0 & \frac{1}{\gamma-1} \end{bmatrix}$$

$$M^{-1^T} = \begin{bmatrix} 1 & -\frac{u_1}{\rho} & -\frac{u_2}{\rho} & \frac{(\gamma-1)u_i u_i}{2} \\ 0 & \frac{1}{\rho} & 0 & -(\gamma-1)u_1 \\ 0 & 0 & \frac{1}{\rho} & -(\gamma-1)u_2 \\ 0 & 0 & 0 & \gamma-1 \end{bmatrix}.$$

The conservative and primitive adjoint operators L and $\tilde{L}$ corresponding to the variations δw and $\delta\tilde{w}$ are then related by

$$\int_{\mathcal{D}} \delta w^T L\psi \; d\mathcal{D}_\xi = \int_{\mathcal{D}} \delta\tilde{w}^T \tilde{L}\psi \; d\mathcal{D}_\xi,$$

with

$$\tilde{L} = M^T L,$$

so that after determining the primitive adjoint operator by direct evaluation of the viscous portion of (6.7), the conservative operator may be obtained by the transformation $L = M^{-1^T}\tilde{L}$. Since the continuity equation contains no viscous terms, it makes no contribution to the viscous adjoint system. Therefore, the derivation proceeds by first examining the adjoint operators arising from the momentum equations.

A.2 Contributions from the Momentum Equations

In order to make use of the summation convention, it is convenient to set $\psi_{j+1} = \phi_j$ for $j = 1, 2$. Then the contribution from the momentum equations is

$$\int_{\mathcal{B}} \phi_k \left(\delta S_{2j}\sigma_{kj} + S_{2j}\delta\sigma_{kj}\right) d\mathcal{B}_\xi - \int_{\mathcal{D}} \frac{\partial\phi_k}{\partial\xi_i}\left(\delta S_{ij}\sigma_{kj} + S_{ij}\delta\sigma_{kj}\right) d\mathcal{D}_\xi. \tag{A.1}$$

The velocity derivatives in the viscous stresses can be expressed as

$$\frac{\partial u_i}{\partial x_j} = \frac{\partial u_i}{\partial\xi_l}\frac{\partial\xi_l}{\partial x_j} = \frac{S_{lj}}{J}\frac{\partial u_i}{\partial\xi_l}$$

with corresponding variations

$$\delta\frac{\partial u_i}{\partial x_j} = \left[\frac{S_{lj}}{J}\right]_I \frac{\partial}{\partial\xi_l}\delta u_i + \left[\frac{\partial u_i}{\partial\xi_l}\right]_{II} \delta\left(\frac{S_{lj}}{J}\right).$$

The variations in the stresses are then

$$\begin{aligned}\delta\sigma_{kj} = \quad & \left\{\mu\left[\tfrac{S_{lj}}{J}\tfrac{\partial}{\partial\xi_l}\delta u_k + \tfrac{S_{lk}}{J}\tfrac{\partial}{\partial\xi_l}\delta u_j\right] + \lambda\left[\delta_{jk}\tfrac{S_{lm}}{J}\tfrac{\partial}{\partial\xi_l}\delta u_m\right]\right\}_I \\ + \quad & \left\{\mu\left[\delta\left(\tfrac{S_{lj}}{J}\right)\tfrac{\partial u_k}{\partial\xi_l} + \delta\left(\tfrac{S_{lk}}{J}\right)\tfrac{\partial u_j}{\partial\xi_l}\right] + \lambda\left[\delta_{jk}\delta\left(\tfrac{S_{lm}}{J}\right)\tfrac{\partial u_m}{\partial\xi_l}\right]\right\}_{II}.\end{aligned}$$

As before, only those terms with subscript I, which contain variations of the flow variables, need be considered further in deriving the adjoint operator. The field contributions that contain δu_i in equation (A.1) appear as

$$-\int_{\mathcal{D}}\frac{\partial\phi_k}{\partial\xi_i}S_{ij}\left\{\mu\left(\frac{S_{lj}}{J}\frac{\partial}{\partial\xi_l}\delta u_k + \frac{S_{lk}}{J}\frac{\partial}{\partial\xi_l}\delta u_j\right) + \lambda\delta_{jk}\frac{S_{lm}}{J}\frac{\partial}{\partial\xi_l}\delta u_m\right\}d\mathcal{D}_\xi.$$

This may be integrated by parts to yield

$$\begin{aligned}& \int_{\mathcal{D}}\delta u_k\frac{\partial}{\partial\xi_l}\left(S_{lj}S_{ij}\frac{\mu}{J}\frac{\partial\phi_k}{\partial\xi_i}\right)d\mathcal{D}_\xi + \int_{\mathcal{D}}\delta u_j\frac{\partial}{\partial\xi_l}\left(S_{lk}S_{ij}\frac{\mu}{J}\frac{\partial\phi_k}{\partial\xi_i}\right)d\mathcal{D}_\xi \\ + \quad & \int_{\mathcal{D}}\delta u_m\frac{\partial}{\partial\xi_l}\left(S_{lm}S_{ij}\frac{\lambda\delta_{jk}}{J}\frac{\partial\phi_k}{\partial\xi_i}\right)d\mathcal{D}_\xi,\end{aligned}$$

where the boundary integral has been eliminated by noting that $\delta u_i = 0$ on the solid boundary. By exchanging indices, the field integrals may be combined to produce

$$\int_{\mathcal{D}}\delta u_k\frac{\partial}{\partial\xi_l}S_{lj}\left\{\mu\left(\frac{S_{ij}}{J}\frac{\partial\phi_k}{\partial\xi_i} + \frac{S_{ik}}{J}\frac{\partial\phi_j}{\partial\xi_i}\right) + \lambda\delta_{jk}\frac{S_{im}}{J}\frac{\partial\phi_m}{\partial\xi_i}\right\}d\mathcal{D}_\xi,$$

which is further simplified by transforming the inner derivatives back to Cartesian coordinates

$$\int_{\mathcal{D}}\delta u_k\frac{\partial}{\partial\xi_l}S_{lj}\left\{\mu\left(\frac{\partial\phi_k}{\partial x_j} + \frac{\partial\phi_j}{\partial x_k}\right) + \lambda\delta_{jk}\frac{\partial\phi_m}{\partial x_m}\right\}d\mathcal{D}_\xi. \tag{A.2}$$

The boundary contributions that contain δu_i in equation (A.1) may be simplified using the fact that

$$\frac{\partial}{\partial\xi_l}\delta u_i = 0 \quad \text{if} \quad l = 1$$

on the boundary $\mathcal{B}$ so that they become

$$\int_{\mathcal{B}} \phi_k S_{2j} \left\{ \mu \left(\frac{S_{2j}}{J} \frac{\partial}{\partial \xi_2} \delta u_k + \frac{S_{2k}}{J} \frac{\partial}{\partial \xi_2} \delta u_j \right) + \lambda \delta_{jk} \frac{S_{2m}}{J} \frac{\partial}{\partial \xi_2} \delta u_m \right\} d\mathcal{B}_\xi. \tag{A.3}$$

Together (A.2) and (A.3) comprise the field and boundary contributions of the momentum equations to the viscous adjoint operator in primitive variables.

A.3 Contributions from the Energy Equation

In order to derive the contribution of the energy equation to the viscous adjoint terms it is convenient to set

$$\psi_4 = \theta, \quad Q_j = u_i \sigma_{ij} + \kappa \frac{\partial}{\partial x_j} \left(\frac{p}{\rho} \right),$$

where the temperature has been written in terms of pressure and density using (2.1.4). The contribution from the energy equation can then be written as

$$\int_{\mathcal{B}} \theta \left(\delta S_{2j} Q_j + S_{2j} \delta Q_j \right) d\mathcal{B}_\xi - \int_{\mathcal{D}} \frac{\partial \theta}{\partial \xi_i} \left(\delta S_{ij} Q_j + S_{ij} \delta Q_j \right) d\mathcal{D}_\xi. \tag{A.4}$$

The field contributions that contain δu_i,δp, and $\delta \rho$ in equation (A.4) appear as

$$\begin{aligned} -\int_{\mathcal{D}} \frac{\partial \theta}{\partial \xi_i} S_{ij} \delta Q_j d\mathcal{D}_\xi = \; & - \int_{\mathcal{D}} \frac{\partial \theta}{\partial \xi_i} S_{ij} \left\{ \delta u_k \sigma_{kj} + u_k \delta \sigma_{kj} \right. \\ & + \left. \kappa \frac{S_{lj}}{J} \frac{\partial}{\partial \xi_l} \left(\frac{\delta p}{\rho} - \frac{p}{\rho} \frac{\delta \rho}{\rho} \right) \right\} d\mathcal{D}_\xi. \end{aligned} \tag{A.5}$$

The term involving $\delta \sigma_{kj}$ may be integrated by parts to produce

$$\int_{\mathcal{D}} \delta u_k \frac{\partial}{\partial \xi_l} S_{lj} \left\{ \mu \left(u_k \frac{\partial \theta}{\partial x_j} + u_j \frac{\partial \theta}{\partial x_k} \right) + \lambda \delta_{jk} u_m \frac{\partial \theta}{\partial x_m} \right\} d\mathcal{D}_\xi,$$

where the conditions $u_i = \delta u_i = 0$ are used to eliminate the boundary integral on $\mathcal{B}$. Notice that the other term in (A.5) that involves δu_k need not be integrated by parts

and is merely carried on as

$$-\int_{\mathcal{D}} \delta u_k \sigma_{kj} S_{ij} \frac{\partial \theta}{\partial \xi_i} d\mathcal{D}_\xi. \tag{A.6}$$

The terms in expression (A.5) that involve δp and $\delta \rho$ may also be integrated by parts to produce both a field and a boundary integral. The field integral becomes

$$\int_{\mathcal{D}} \left(\frac{\delta p}{\rho} - \frac{p}{\rho} \frac{\delta \rho}{\rho} \right) \frac{\partial}{\partial \xi_l} \left(S_{lj} S_{ij} \frac{\kappa}{J} \frac{\partial \theta}{\partial \xi_i} \right) d\mathcal{D}_\xi$$

which may be simplified by transforming the inner derivative to Cartesian coordinates

$$\int_{\mathcal{D}} \left(\frac{\delta p}{\rho} - \frac{p}{\rho} \frac{\delta \rho}{\rho} \right) \frac{\partial}{\partial \xi_l} \left(S_{lj} \kappa \frac{\partial \theta}{\partial x_j} \right) d\mathcal{D}_\xi. \tag{A.7}$$

The boundary integral becomes

$$\int_{\mathcal{B}} \kappa \left(\frac{\delta p}{\rho} - \frac{p}{\rho} \frac{\delta \rho}{\rho} \right) \frac{S_{2j} S_{ij}}{J} \frac{\partial \theta}{\partial \xi_i} d\mathcal{B}_\xi. \tag{A.8}$$

This can be simplified by transforming the inner derivative to Cartesian coordinates

$$\int_{\mathcal{B}} \kappa \left(\frac{\delta p}{\rho} - \frac{p}{\rho} \frac{\delta \rho}{\rho} \right) \frac{S_{2j}}{J} \frac{\partial \theta}{\partial x_j} d\mathcal{B}_\xi, \tag{A.9}$$

and identifying the normal derivative at the wall

$$\frac{\partial}{\partial n} = S_{2j} \frac{\partial}{\partial x_j}, \tag{A.10}$$

and the variation in temperature

$$\delta T = \frac{1}{R} \left(\frac{\delta p}{\rho} - \frac{p}{\rho} \frac{\delta \rho}{\rho} \right),$$

to produce the boundary contribution

$$\int_{\mathcal{B}} k \delta T \frac{\partial \theta}{\partial n} d\mathcal{B}_\xi. \tag{A.11}$$

This term vanishes if T is constant on the wall but persists if the wall is adiabatic.

There is also a boundary contribution left over from the first integration by parts (A.4) which has the form

$$\int_{\mathcal{B}} \theta\delta\left(S_{2j}Q_j\right) d\mathcal{B}_\xi, \tag{A.12}$$

where

$$Q_j = k\frac{\partial T}{\partial x_j},$$

since $u_i = 0$. Notice that for future convenience in discussing the adjoint boundary conditions resulting from the energy equation, both the δw and δS terms corresponding to subscript classes I and II are considered simultaneously. If the wall is adiabatic

$$\frac{\partial T}{\partial n} = 0,$$

so that using (A.10),

$$\delta\left(S_{2j}Q_j\right) = 0,$$

and both the δw and δS boundary contributions vanish.

On the other hand, if T is constant $\frac{\partial T}{\partial \xi_l} = 0$ for $l = 1$, so that

$$Q_j = k\frac{\partial T}{\partial x_j} = k\left(\frac{S_l j}{J}\frac{\partial T}{\partial \xi_l}\right) = k\left(\frac{S_{2j}}{J}\frac{\partial T}{\partial \xi_2}\right).$$

Thus, the boundary integral (A.12) becomes

$$\int_{\mathcal{B}} k\theta\left\{\frac{{S_{2j}}^2}{J}\frac{\partial}{\partial \xi_2}\delta T + \delta\left(\frac{{S_{2j}}^2}{J}\right)\frac{\partial T}{\partial \xi_2}\right\} d\mathcal{B}_\xi\ . \tag{A.13}$$

Therefore, for constant T, the first term corresponding to variations in the flow field contributes to the adjoint boundary operator and the second set of terms corresponding to metric variations contribute to the cost function gradient.

All together, the contributions from the energy equation to the viscous adjoint operator are the three field terms (A.6), (A.6) and (A.7), and either of two boundary contributions (A.11) or (A.13), depending on whether the wall is adiabatic or has constant temperature.

A.4 The Viscous Adjoint Field Operator

Collecting together the contributions from the momentum and energy equations, the viscous adjoint operator in primitive variables can be expressed as

$$\begin{aligned}
(\tilde{L}\psi)_1 = &- \frac{p}{\rho^2}\frac{\partial}{\partial \xi_l}\left(S_{lj}\kappa\frac{\partial \theta}{\partial x_j}\right) \\
(\tilde{L}\psi)_{i+1} = & \frac{\partial}{\partial \xi_l}\left\{S_{lj}\left[\mu\left(\frac{\partial \phi_i}{\partial x_j}+\frac{\partial \phi_j}{\partial x_i}\right)+\lambda\delta_{ij}\frac{\partial \phi_k}{\partial x_k}\right]\right\} \\
&+ \frac{\partial}{\partial \xi_l}\left\{S_{lj}\left[\mu\left(u_i\frac{\partial \theta}{\partial x_j}+u_j\frac{\partial \theta}{\partial x_i}\right)+\lambda\delta_{ij}u_k\frac{\partial \theta}{\partial x_k}\right]\right\} \\
&- \sigma_{ij}S_{lj}\frac{\partial \theta}{\partial \xi_l} \qquad \text{for} \quad i=1,2 \\
(\tilde{L}\psi)_4 = & \frac{1}{\rho}\frac{\partial}{\partial \xi_l}\left(S_{lj}\kappa\frac{\partial \theta}{\partial x_j}\right).
\end{aligned}$$

The conservative viscous adjoint operator may now be obtained by the transformation

$$L = M^{-1^T}\tilde{L}.$$

A.5 Viscous Adjoint Boundary Conditions

In this section, the viscous adjoint boundary condition for both inverse design and drag minimization are discussed.

A.5.1 Inverse Design

In the continuous adjoint case, the boundary term that arises from the momentum equations including both the δw and δS components (A.1) takes the form

$$\int_{\mathcal{B}} \phi_k \delta\left(S_{2j}\sigma_{kj}\right) d\mathcal{B}_\xi.$$

Replacing the metric term with the corresponding local face area S_2 and unit normal n_j defined by

$$|S_2| = \sqrt{S_{2j}S_{2j}}, \quad n_j = \frac{S_{2j}}{|S_2|}$$

then leads to

$$\int_{\mathcal{B}} \phi_k \delta \left(|S_2| \, n_j \sigma_{kj} \right) d\mathcal{B}_\xi.$$

Defining the components of the surface stress as

$$\tau_k = n_j \sigma_{kj}$$

and the physical surface element

$$dS = |S_2| \, d\mathcal{B}_\xi,$$

the integral may then be split into two components

$$\int_{\mathcal{B}} \phi_k \tau_k \, |\delta S_2| \, d\mathcal{B}_\xi + \int_{\mathcal{B}} \phi_k \delta \tau_k dS, \tag{A.14}$$

where only the second term contains variations in the flow variables and must consequently cancel the δw terms arising in the cost function. The first term will appear in the expression for the gradient.

A general expression for the cost function that allows cancellation with terms containing $\delta\tau_k$ has the form

$$I = \int_{\mathcal{B}} \mathcal{N}(\tau) dS, \tag{A.15}$$

corresponding to a variation

$$\delta I = \int_{\mathcal{B}} \frac{\partial \mathcal{N}}{\partial \tau_k} \delta \tau_k dS,$$

for which cancellation is achieved by the adjoint boundary condition

$$\phi_k = \frac{\partial \mathcal{N}}{\partial \tau_k}.$$

Natural choices for $\mathcal{N}$ arise from force optimization and as measures of the deviation

of the surface stresses from desired target values.

In the inverse design case, in order to control the surface pressure and normal stress one can measure the difference

$$n_j \left\{ \sigma_{kj} + \delta_{kj} \left(p - p_d \right) \right\},$$

where p_d is the desired pressure. The normal component is then

$$\tau_n = n_k n_j \sigma_{kj} + p - p_d,$$

so that the measure becomes

$$\begin{aligned} \mathcal{N}(\tau) &= \frac{1}{2} \tau_n^2 \\ &= \frac{1}{2} n_l n_m n_k n_j \left\{ \sigma_{lm} + \delta_{lm} \left(p - p_d \right) \right\} \cdot \left\{ \sigma_{kj} + \delta_{kj} \left(p - p_d \right) \right\}. \end{aligned}$$

Defining the viscous normal stress as

$$\tau_{vn} = n_k n_j \sigma_{kj},$$

the measure can be expanded as

$$\begin{aligned} \mathcal{N}(\tau) &= \frac{1}{2} n_l n_m n_k n_j \sigma_{lm} \sigma_{kj} + \frac{1}{2} \left(n_k n_j \sigma_{kj} + n_l n_m \sigma_{lm} \right) \left(p - p_d \right) + \frac{1}{2} \left(p - p_d \right)^2 \\ &= \frac{1}{2} \tau_{vn}^2 + \tau_{vn} \left(p - p_d \right) + \frac{1}{2} \left(p - p_d \right)^2. \end{aligned}$$

For cancellation of the boundary terms

$$\phi_k \left(n_j \delta \sigma_{kj} + n_k \delta p \right) = \left\{ n_l n_m \sigma_{lm} + n_l^2 \left(p - p_d \right) \right\} n_k \left(n_j \delta \sigma_{kj} + n_k \delta p \right) \tag{A.16}$$

leading to the boundary condition

$$\phi_k = n_k \left(\tau_{vn} + p - p_d \right).$$

In the case of high Reynolds number, this is well approximated by the equations

$$\phi_k = n_k \left(p - p_d \right), \tag{A.17}$$

which should be compared with the single scalar equation derived for the inviscid boundary condition. In the case of an inviscid flow, choosing

$$\mathcal{N}(\tau) = \frac{1}{2} \left(p - p_d \right)^2$$

requires

$$\phi_k n_k \delta p = \left(p - p_d \right) n_k^2 \delta p = \left(p - p_d \right) \delta p \tag{A.18}$$

which is satisfied by equation (A.17), but which represents an over-specification of the boundary condition since only the single condition need be specified to ensure cancellation.

A.5.2 Drag Minimization

Pressure Drag Minimization

In the continuous adjoint case, if the drag is to be minimized, then the cost function is the drag coefficient,

$$\begin{aligned} I &= C_d \\ &= \left(\frac{1}{c} \int_{B_W} C_p \frac{\partial y}{\partial \xi} d\xi \right) \cos\alpha + \left(\frac{1}{c} \int_{B_W} -C_p \frac{\partial x}{\partial \xi} d\xi \right) \sin\alpha. \end{aligned}$$

A variation in the shape causes a variation ∂p in the pressure and consequently a variation in the cost function,

$$\begin{aligned}\delta I = \quad & \frac{1}{c}\int_{B_W} C_p \left(\frac{\partial y}{\partial \xi}\cos\alpha - \frac{\partial x}{\partial \xi}\sin\alpha\right)\partial p d\xi \\ + \quad & \frac{1}{c}\int_{B_W} C_p \left(\delta\left(\frac{\partial y}{\partial \xi}\right)\cos\alpha - \delta\left(\frac{\partial x}{\partial \xi}\right)\sin\alpha\right) d\xi.\end{aligned} \tag{A.19}$$

As in the inverse design case, the first term is a function of the state vector, and therefore is incorporated into the boundary condition, where the integrand replaces the pressure difference term in equation (A.18) producing the following boundary condition,

$$\phi_k n_k = -\frac{1}{\frac{1}{2}\gamma P_\infty M_\infty^2}\begin{bmatrix}\cos\alpha \\ \sin\alpha\end{bmatrix} n_k. \tag{A.20}$$

Skin Friction Drag Minimization

For viscous force optimization, the cost function should measure skin friction drag. The skin friction force in the x_i direction is

$$CD_{fi} = \int_{\mathcal{B}} \sigma_{ij} dS_j = \int_{\mathcal{B}} S_{2j}\sigma_{ij} d\mathcal{B}_\xi$$

so that the force in a direction with cosines l_i has the form

$$C_{nf} = \int_{\mathcal{B}} l_i S_{2j}\sigma_{ij} d\mathcal{B}_\xi.$$

Expressed in terms of the surface stress τ_i, this corresponds to

$$C_{nf} = \int_{\mathcal{B}} l_i \tau_i dS,$$

so that basing the cost function (A.15) on this quantity gives

$$\mathcal{N} = l_i \tau_i.$$

Cancellation with the flow variation terms in equation (A.14) therefore mandates the continuous viscous adjoint boundary condition

$$\phi_k = l_k \tag{A.21}$$

where,

$$l_k = -\frac{1}{\frac{1}{2}\gamma P_\infty M_\infty^2}\begin{bmatrix}\cos\alpha \\ \sin\alpha\end{bmatrix}.$$

If one would take the dot product of equation (A.21) then the resulting equation is identical to equation (A.20). Therefore, the continuous adjoint boundary condition for skin friction drag minimization also satisfies the pressure drag minimization cost function, and thus the two boundary conditions are inseparable. This choice of boundary condition also eliminates the first term in equation (A.14) so that it need not be included in the gradient calculation. Notice that the choice for the first and fourth Lagrange multipliers can be arbitrarily set to zero or a zeroth order extrapolation across the wall can be adopted since equation (A.21) provides no suggestion for these values. The effect of this boundary condition is explored in the Results section.

Total Drag Minimization

Since the continuous adjoint boundary condition for skin friction drag minimization also satisfies the pressure drag minimization cost function, then equation (A.21) is used for total drag minimization.

Bibliography

[1] J. J. Alonso. *Parallel Computation of Unsteady and Aeroelastic Flows Using an Implicit Multigrid-Driven Algorithm*. PhD thesis, Princeton University, Dept. of Mechanical and Aerospace Engineering, February 1997.

[2] J.J. Alonso, A. Jameson, and I. Kroo. Advanced algorithms for design and optimization of quiet supersonic platforms. *AIAA paper 02-0144*, 40th. Aerospace Sciences Meeting and Exhibit, Reno, Nevada, January 2002.

[3] W. K. Anderson and V. Venkatakrishnan. Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation. *AIAA paper 97-0643*, 35th Aerospace Sciences Meeting and Exhibit, Reno, Nevada, January 1997.

[4] B. S. Baldwin and H. Lomax. Thin layer approximation and algebraic model for separated flows. *AIAA paper 78-275*, 1978.

[5] F. Bauer, P. Garabedian, D. Korn, and A. Jameson. *Supercritical Wing Sections II*. Springer Verlag, New York, 1975.

[6] Charles S. Beightler. *Foundations of Optimization*. Prentice-Hall, New Jersey, 1979. Edited by W. J. Fabrycky and J. H. Mize.

[7] J. Boussinesq. Essai sur la theorie des eaux courantes. *Mem. Presentes Acad. Sci*, 23:46, 1877.

[8] A. Brandt. Multilevel adaptive technique for fast numerical solution to boundary value problems. *Proc. 3rd. International Conference on Numerical Methods in Fluid Dynamics, Lecture Notes in Physics*, 18:82–9, 1972.

[9] A. Brandt. Multi-level adaptive solutions to boundary value problems. *Math. Comp.*, 31:333–390, 1977.

[10] G. W. Burgreen and O. Baysal. Three-dimensional aerodynamic shape optimization of wings using sensitivity analysis. *AIAA paper 94-0094*, 32nd Aerospace Sciences Meeting and Exhibit, Reno, Nevada, January 1994.

[11] R. Campbell and L. Smith. A hybrid algorithm for transonic airfoil and wing design. *AIAA paper 87-2552-CP*, Aerospace Science and Exhibition, Reno, NV, 1987.

[12] S. E. Cliff, J. .J. Reuther, D. A. Saunders, and R. M. Hicks. Single-point and multipoint aerodynamic shape optimization of high-speed civil transport. *Journal of Aircraft*, 38(6):997–1005, November-December 2001.

[13] L. Dadone. Personal communications on advanced rotor blade profiles. July 2002.

[14] Dahlquist and R. Jeltsch. Stability on the imaginary axis and a-stability of linear multistep methods. *BIT*, 18:170–174, 1978.

[15] S. S. Davis. NACA 64A010 (NASA Ames model) oscillatory pitching. AGARD Report 702, August 1982. In Compendium of Unsteady Aerodynamic Measurements.

[16] Van der Houwen. *Construction of Integration Formulas for Initial Value Problems*. North-Holland, Amsterdam, Holland, 1977.

[17] J. Elliot and J. Peraire. Aerodynamic design using unstructured meshes. *AIAA paper 96-1941*, 1996.

[18] R. P. Fedorenko. A relaxation method for solving elliptic differential equations. *USSR Computational Math. and Math. Physics*, 1:1092–6, 1962.

[19] R.P. Fedorenko. The speed of convergence of one iterative process. *USSR Comp. Math. and Math. Phys.*, 4:227–235, 1964.

[20] R. Florea and K. C. Hall. Sensitivity analysis of unsteady inviscid flow through turbomachinery cascades. *AIAA paper 2000-0130*, 38th Aerospace Sciences Meeting and Exhibit, Reno, Nevada, January 2000.

[21] P. R. Garabedian and D. G. Korn. Numerical design of transonic airfoils. In B. Hubbard, editor, *Proceedings of SYNSPADE 1970*, pages 253–271, Academic Press, New York, 1971.

[22] C. W. Gear. *Numerical Initial Value Problems*. Prentice Hall, Englewood Cliffs, 1971.

[23] K. Ghayour and O. Baysal. Unsteady aerodynamics and shape optimization using modified transonic small disturbance equation. *AIAA paper 99-0654*, 37th Aerospace Sciences Meeting and Exhibit, Reno, Nevada, January 1999.

[24] P.E. Gill, W. Murray, and M.A. Saunders. User's guide for SNOPT (version 5.3). a FORTRAN package linear and nonlinear programming. Technical report, Stanford University, Department of Operations Research, 2002.

[25] P.E. Gill, W. Murray, M.A. Saunders, and M.A. Wright. User's guide for NPSOL (version 5.0). a FORTRAN package nonlinear programming. Technical report, Stanford University, Department of Operations Research, 1995.

[26] R. M. Hicks and P. A. Henne. Wing design by numerical optimization. *Journal of Aircraft*, 15:407–412, 1978.

[27] R. M. Hicks and P. A. Henne. Wing design by numerical optimization. *AIAA paper 79-0080*, 1979.

[28] C. Hirsch. *Numerical Computation of Internal and External Flows*, volume 1. John Wiley & Sons, New York, 1988.

[29] A. Iollo, M. Salas, and S. Ta'asan. Shape optimization governed by the euler equations using an adjoint method. *Nasa Contractor Report 93-78*, ICASE report, Langley Research Center, November 1993.

[30] A. Jameson. Stanford, CA. Private communication, January 2003.

[31] A. Jameson. Solution of the Euler equations for two dimensional transonic flow by a multigrid method. *Applied Mathematics and Computations*, 13:327–356, 1983.

[32] A. Jameson. Multigrid algorithms for compressible flow calculations. In W. Hackbusch and U. Trottenberg, editors, *Lecture Notes in Mathematics, Vol. 1228*, pages 166–201. Proceedings of the 2nd European Conference on Multigrid Methods, Cologne, 1985, Springer-Verlag, 1986.

[33] A. Jameson. Aerodynamic design via control theory. *Journal of Scientific Computing, also ICASE Report No. 88-64*, 3:233–260, 1988.

[34] A. Jameson. Computational aerodynamics for aircraft design. *Science*, 245:361–371, 1989.

[35] A. Jameson. Automatic design of transonic airfoils to reduce the shock induced pressure drag. In *Proceedings of the 31st Israel Annual Conference on Aviation and Aeronautics, Tel Aviv*, pages 5–17, February 1990.

[36] A. Jameson. Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings. *AIAA paper 91-1596*, AIAA 10th Computational Fluid Dynamics Conference, Honolulu, Hawaii, June 1991.

[37] A. Jameson. Optimum aerodynamic design using CFD and control theory. AIAA Paper 95-1729, San Diego, CA, 1995.

[38] A. Jameson. Optimum aerodynamic design using CFD and control theory. *AIAA paper 95-1729*, AIAA 12th Computational Fluid Dynamics Conference, San Diego, CA, June 1995.

[39] A. Jameson, J. J. Alonso, J. Reuther, L. Martinelli, and J. Vassberg. Aerodynamic shape optimization techniques based on control theory. *AIAA paper 98-2538*, AIAA 29th. Fluid Dynamics Conference, Albuquerque, NM, June 1998.

[40] A. Jameson and T.J. Baker. Solution of the Euler equations for complex configurations. *AIAA paper 83-1929*, AIAA 6th Computational Fluid Dynamics Conference, Danvers, MA, July 1983.

[41] A. Jameson, L. Martinelli, and N. Pierce. Optimum aerodynamic design using the navier-stokes equations. *AIAA paper 97-0101*, 35th Aerospace Sciences Meeting and Exhibit, RENO, NEVADA, JANUARY 1997.

[42] A. Jameson, L. Martinelli, and N. Pierce. Optimum aerodynamic design using the navier-stokes equations. *Theoretical Computational Fluid Dynamics*, 10:213–237, 1998.

[43] A. Jameson, W. Schmidt, and E. Turkel. Numerical solutions of the Euler equations by finite volume methods with Runge-Kutta time stepping schemes. *AIAA paper 81-1259*, January 1981.

[44] A. Jameson and J. C. Vassberg. Studies of alternative numerical optimization methods applied to the brachistochrone problem. *OptiCON 99*, 1999.

[45] R. E. Kielb. Cfd for turbomachinery unsteady flows - an aeroelastic design perspective. *AIAA paper 2001-0429*, 2001.

[46] S. Kim. Design optimization of high-lift configurations using a viscous adjoint-based method. Technical report.

[47] S. Kim, J.J. Alonso, and A. Jameson. A gradient accuracy study for the adjoint-based navier-stokes design method. *AIAA paper 99-0299*, AIAA 37th Aerospace Sciences Meeting and Exhibit, Reno, NV, January 1999.

[48] J. D. Lambert. *Numerical Methods for Ordinary Differential Systems: the Initial Value Problem*. Wiley, Chichester, NY, 1991.

[49] H. W. Liepmann and A. Roshko. *Elements of gasdynamics*. Wiley, New York, 1957.

[50] M. J. Lighthill. A new method of two-dimensional aerodynamic design. *R & M 1111*, Aeronautical Research Council, 1945.

[51] J. L. Lions. *Optimal Control of Systems Governed by Partial Differential Equations*. Springer-Verlag, New York, 1971. Translated by S.K. Mitter.

[52] J. N. Lyness and C. B. Moler. Numerical differentiation of analytic functions. *SIAM Journal on Numerical Analysis*, 4(2):202–210, June 1967.

[53] L. Martinelli. *Calculations of Viscous Flow with a Multigrid Method*. PhD thesis, Princeton University, Princeton, NJ, October 1987.

[54] J. R. R. A. Martins. *A Coupled-Adjoint Method for High-Fidelity Aero-Structural Optimization*. Ph.d. dissertation, Department of Aeronautics and Astronautics, Stanford University, Stanford, CA, November 2002.

[55] J. R. R. A. Martins, I.M. Kroo, and J.J. Alonso. An automated method for sensitivity analysis using complex variables. *AIAA paper 2000-0689*, 38th Aerospace Sciences Meeting and Exhibit, Reno, Nevada, January 2000.

[56] G. B. McFadden. An artificial viscosity method for the design of supercritical airfoils. *Internal report, and Ph.D. Thesis C00-3077-158*, New York University, 1979.

[57] Matthew McMullen. *The Application of Non-Linear Frequency Domain Methods to the Euler and Navier-Stokes Equations*. Ph.d. dissertation, Department of Aeronautics and Astronautics, Stanford University, Stanford, CA, March 2003.

[58] N. D. Melson, M. D. Sanetrik, and H. L. Atkins. Time-accurate Navier-Stokes calculations with multigrid acceleration. Copper Mountain, CO, July 1993. Proceedings of the Sixth Copper Mountain Conference on Multigrid Methods.

[59] S. Nadarajah and A. Jameson. A comparison of the continuous and discrete adjoint approach to automatic aerodynamic optimization. *AIAA paper 2000-0667*, 38th Aerospace Sciences Meeting and Exhibit, Reno, Nevada, January 2000.

[60] S. Nadarajah and A. Jameson. Studies of the continuous and discrete adjoint approaches to viscous automatic aerodynamic shape optimization. *AIAA paper 2001-2530*, 15th. Computational Fluid Dynamics Conference, Anaheim, CA, June 2001.

[61] S. Nadarajah and A. Jameson. Optimal control of unsteady flows using a time accurate method. *AIAA paper 2002-5436*, 9th. AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization Conference, Atlanta, GA, September 2002.

[62] S. Nadarajah, A. Jameson, and J. J. Alonso. An adjoint method for the calculation of non-collocated sensitivities in supersonic flow. In *Proceedings of First MIT Conference on Computational Fluid and Solid Mechanics*, Cambridge, MA, June 2001.

[63] S. Nadarajah, A. Jameson, and J. J. Alonso. An adjoint method for the calculation of remote sensitivities in supersonic flow. *AIAA paper 2002-0261*, 40th. Aerospace Sciences Meething and Exhibit, Reno, NV, Jan 2002.

[64] S. Nadarajah, A. Jameson, and J. J. Alonso. Sonic boom reduction using an adjoint method for wing-body configurations in supersonic flow. *AIAA paper 2002-5547*, 9th. AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization Conference, Atlanta, GA, September 2002.

[65] S. Nadarajah, S. Kim, A. Jameson, and J. J. Alonso. Sonic boom reduction using an adjoint method for supersonic transport aircraft configuration. Technical report, IUTAM Symposium Transsonicum IV, Gottingen, Germany, September 2002.

[66] J. Von Neumann and R. D. Richtmyer. A method for the numerical calculations of hydrodynamical shocks. *Journal Mathematical Physics*, 21, 1950.

[67] Committee on Breakthrough Technology for Commercial Supersonic Aircraft. Commercial supersonic technology: The way ahead. Technical report, National Research Council, National Academy Press, Washington DC, March 2001.

[68] O. Pironneau. *Optimal Shape Design for Elliptic Systems*. Springer-Verlag, New York, 1984.

[69] J. Reuther and A. Jameson. Aerodynamic shape optimization of wing and wing-body configurations using control theory. *AIAA paper 95-0123*, 33rd Aerospace Sciences Meeting and Exhibit, Reno, Nevada, January 1995.

[70] J. Reuther, A. Jameson, J. J. Alonso, M. J. Rimlinger, and D. Sauders. Constrained multipoint aerodynamic shape optimization using and adjoint formulation and parallel computers. *AIAA paper 97-0103*, AIAA 35th Aerospace Sciences Meeting, Reno, NV, January 1997.

[71] J. Reuther, A. Jameson, J. J. Alonso, M. J. Rimlinger, and D. Saunders. Constrained multipoint aerodynamic shape optimization using an adjoint formulation and parallel computers. *AIAA paper 97-0103*, 35th Aerospace Sciences Meeting and Exhibit, Reno, Nevada, January 1997.

[72] J. J. Reuther. Aerodynamic shape optimization using control theory. *Ph. D. Dissertation*, University of California, Davis, Davis, CA, June 1996.

[73] J. J. Reuther, J. J. Alonso, M. Rimlinger, and A. Jameson. Aerodynamic shape optimization of supersonic aircraft configurations via an adjoint formulation on distributed memory parallel computers. Bellevue, WA, September 1996. 6th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization. AIAA Paper 96-4045.

[74] A. Rizzi and H. Viviand. Numerical methods for the computation of inviscid transonic flows with shock waves. In *Proc. GAMM Workshop*, Stockholm, 1979.

[75] D. Saunders. Bilinear and trilinear interpolation routines. Technical report, Sterling Software/NASA Ames, Moffett Field, CA, 1998.

[76] R. Seebass and B. Argrow. Sonic boom minimization revisited. *AIAA paper 98-2956*, 2nd AIAA Theoretical Fluid Mechanics Meeting, Albuquerque, NM, June 1998.

[77] G. R. Shubin and P. D. Frank. A comparison of the implicit gradient approach and the variational approach to aerodynamic design optimization. *internal paper AMS-TR-164*, Boeing Computer Services, April 1991.

[78] S. Ta'asan, G. Kuruvila, and M. D. Salas. Aerodynamic design and optimization in one shot. *AIAA paper 92-0025*, 30th Aerospace Sciences Meeting and Exhibit, Reno, Nevada, January 1992.

[79] C. Thomas. Extrapolation of sonic boom pressure signatures by the waveform parameter method. *NASA TN D-6832*, 1972.

[80] G. B. Whitham. The flow pattern of a supersonic projectile. *Commun. Pure Appl. Math*, V(3):301–348, 1952.

[81] P. G. Wilby. Shockwaves in the rotor world – a personal perspective of 30 years of rotor aerodynamic developments in the uk. Technical report, March 1998.

[82] K. Yee, Y. Kim, and D. Lee. Aerodynamic shape optimization of rotor airfoils undergoing unsteady motion. *AIAA paper 99-3107*, 37th Aerospace Sciences Meeting and Exhibit, Reno, Nevada, January 1999.